



Calhoun: The NPS Institutional Archive
DSpace Repository

Theses and Dissertations

Thesis and Dissertation Collection

1976

Reduced order approximations to higher order linear systems.

Thompson, Jerry Dennis

<http://hdl.handle.net/10945/17922>

Downloaded from NPS Archive: Calhoun



Calhoun is a project of the Dudley Knox Library at NPS, furthering the precepts and goals of open government and government transparency. All information contained herein has been approved for release by the NPS Public Affairs Officer.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

<http://www.nps.edu/library>

REDUCED ORDER APPROXIMATIONS
TO
HIGHER ORDER LINEAR SYSTEMS

Jerry Dennis Thompson

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

REDUCED ORDER APPROXIMATIONS
TO
HIGHER ORDER LINEAR SYSTEMS

by

Jerry Dennis Thompson

Thesis Advisor:

A. Gerba Jr.

Approved for public release; distribution unlimited.

T174145

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) REDUCED ORDER APPROXIMATIONS TO HIGHER ORDER LINEAR SYSTEMS		5. TYPE OF REPORT & PERIOD COVERED Masters Thesis; June 1976
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Jerry Dennis Thompson		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE June 1976
		13. NUMBER OF PAGES 79
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Reduced Linear Approximations Approximation methods Reduced Equations Computer Approximations		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Low order models are derived by a computer program technique which utilizes the Routh Approximation Method of analysis. Comparisons are made between this method and that of the Dominant Pole Method and the Iterative Optimization Method of analysis. Low order models are developed from higher-order		

linear systems and compared to that system in response to input excitations consisting of a Step and a Ramp.

Graphical displays and numerical tables provide a basis for error analysis and comparisons between the approximation techniques.

REDUCED ORDER APPROXIMATIONS TO HIGHER ORDER LINEAR SYSTEMS

by

ennis
Jerry D. Thompson
Lieutenant, United States Navy
B.S., University of New Mexico, 1971

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the
NAVAL POSTGRADUATE SCHOOL
June 1976

ABSTRACT

Low order models are derived by a computer program technique which utilizes the Routh Approximation Method of analysis. Comparisons are made between this method and that of the Dominant Pole Method and the Iterative Optimization Method of analysis.

Low order models are developed from higher-order, linear systems and compared to that system in response to input excitations consisting of a Step and a Ramp.

Graphical displays and numerical tables provide a basis for error analysis and comparisons between the approximation techniques.

TABLE OF CONTENTS

I.	INTRODUCTION-----	10
II.	NATURE OF THE PROBLEM-----	12
III.	COMPUTER PROGRAM CRITERIA-----	14
	A. GENERAL-----	14
	B. SUBROUTINES-----	15
IV.	FOURTH ORDER EXAMPLE-----	16
	A. THREE STEP PROCEDURE-----	16
V.	COMPUTER EXAMPLE-----	21
	A. SEVENTH ORDER SYSTEM-----	21
VI.	COMPARISONS TO OTHER METHODS-----	29
	A. DOMINANT POLE METHOD-----	29
	B. ITERATIVE OPTIMIZATION METHOD-----	30
	C. PRESENTATION OF DATA-----	31
VII.	CONCLUSION-----	45
	COMPUTER OUTPUT-----	46
	COMPUTER PROGRAM-----	56
	BIBLIOGRAPHY-----	78
	INITIAL DISTRIBUTION LIST-----	79

LIST OF TABLES

Table		Page
IV.1	Routh table of Alpha and Beta computations.....	19
IV.2	Poles and zeros of fourth order Routh Approximations to the fourth order example.....	20
V.1	Poles and zeros of seventh order Routh Approximations to the seventh order example.....	22
VI.1	Performance measure comparison data.....	44

LIST OF FIGURES

Figure		Page
5.1	Routh second-order approximant versus seventh-order system (Step Input).....	23
5.2	Routh third-order approximant versus seventh-order system (Step Input).....	24
5.3	Routh fourth-order approximant versus seventh-order system (Step Input).....	25
5.4	Routh second-order approximant versus seventh-order system (Ramp Input).....	26
5.5	Routh third-order approximant versus seventh-order system (Ramp Input).....	27
5.6	Routh fourth-order approximant versus seventh-order system (Ramp Input).....	28
6.1	Dominant Pole second-order approximant versus seventh-order system (Step Input).....	32
6.2	Dominant Pole third-order approximant versus seventh-order system (Step Input).....	33
6.3	Dominant Pole fourth-order approximant versus seventh-order system (Step Input).....	34
6.4	Dominant Pole second-order approximant versus Routh second-order approximant (Step Input).....	35
6.5	Dominant Pole third-order approximant versus Routh third-order approximant (Step Input).....	36
6.6	Dominant Pole fourth-order approximant versus Routh fourth-order approximant (Step Input).....	37

6.7	Optimum Minimization second order approximant versus seventh-order system (Step Input).....	38
6.8	Optimum Minimization third-order approximant versus seventh-order system (Step Input).....	39
6.9	Optimum Minimization fourth-order approximant versus seventh-order system (Step Input).....	40
6.10	Optimum Minimization second-order approximant versus Routh second-order approximant (Step Input).....	41
6.11	Optimum Minimization third-order approximant versus Routh third-order approximant (Step Input).....	42
6.12	Optimum Minimization fourth-order approximant versus Routh fourth-order approximant (Step Input).....	43

TABLE OF SYMBOLS

<u>Symbol</u>	<u>Definition</u>
A	Original seventh-order equation
R	Routh approximation
D	Dominant Pole approximation
O	Optimum minimization equation
M_{pt}	Peak overshoot = Y_{max}/Y_{ss}
T_d	Delay time
T_r	Rise time
T_s	Settling time
E	Error=A-Approximant (response)
J	Average absolute error
C_d	Denominator coefficients (Routh Table)
C_n	Numerator coefficients (Routh Table)

ACKNOWLEDGEMENT

The author wishes to express sincere appreciation and gratitude to Professors A. Gerba Jr. and George J. Thaler of the Electrical Engineering Department, Naval Postgraduate School of Monterey, California, for their advice and guidance.

I. INTRODUCTION

The complexity involved in designing a control system of reasonably low order for dynamic systems of substantially higher order is a subject about which much has been written. The list of references at the end of this study include only a small portion of the material available concerning this subject. A new development has been introduced by Maurice F. Hutton and Bernard Friedland, [Ref 1], which provides a systematic and analytical approach to obtaining reduced-order approximations from higher-order transfer functions. This method is referred to as the "Routh Approximation Method," and is based on an expansion that uses the Routh table of the original higher-order transfer function. The desire for low order models to simulate higher-order practical systems, such as electrical power plants, chemical processes, aircraft model designs and electronic circuitry is obvious when dealing with such complex systems. The "Routh Approximation Method", is thus, an extremely powerful tool to utilize in dealing with these types of problems.

In comparison to other available methods presently used, the Routh method has many advantages. The widely used "Dominant Pole", approximation method, which is based on approximating a system by utilizing the poles, or characteristic roots nearest the imaginary axis, has the disadvantage that the roots of the characteristic equation must be found. For a very-high-order system, this is not a trivial task. The "Pad'e" approximation is based on setting the numerator and denominator orders to a desired value, the coefficients are then chosen so that the Taylor series expansions of the approximant and the original transfer

function agree in as many terms as possible. This method produces accurate results, however, it is limited in application to single-input, single-output system analysis. In addition, an unstable approximant may be obtained from a stable system since the approximant's poles depend on both the original equation's numerator and denominator.

The "Routh Approximation Method," preserves stability if the original transfer function is stable. It provides an efficient means of obtaining lower-order approximants for multiple inputs or outputs and is very adaptable to computer programming.

This thesis is concerned with the development of a computer program which utilizes the "Routh Approximation Method," to obtain lower-order transfer functions from higher-order functions and compares the resulting equations to the original equations by graphically displaying the response of each to various input excitations. The program was designed to provide output data which is useful in determining which degree of approximant is best suited to simulate the original higher-order equation. The orders available are the first through the fourth order reduced transfer functions. The program is capable of reducing transfer functions up to, and including the tenth order and, without loss of generality, may be extended to handle an almost unlimited order.

For illustrative purposes, a seventh order system was used, as an example, to demonstrate the computer program's capabilities. This particular system was chosen merely to demonstrate the simplicity involved in utilizing the program and has no physical relationship to any real system.

II. NATURE OF THE PROBLEM

In approximating higher-order systems by lower-order models, a linear system is desirable since the linear characteristic equations are less complicated than the non-linear equations. Therefore, this study is primarily concerned with linear, time invariant characteristic equations, since the objective in developing the computer program was to illustrate a simple analytical approach to obtaining lower order characteristic equations from higher order systems.

It is obvious that a reduced-order model cannot characterize a given system as accurately as a higher-order model. The validity of the lower-order model is based upon its degree of success in approximating the higher order system in representing the characteristics of primary interest.

Interpreting the solution of a higher-order system often results in computational difficulties which are reduced by appropriate selection of a reduced-order approximation.

Ideally, a reduced-order model would approximate the higher-order system in both low and high frequency ranges. In doing so, some accuracy is lost in compensating for the different responses of the system to variations in frequency.

The low frequency model more closely approximates the higher-order system than a model composed of both low and high order frequency characteristics. The procedure

utilized in this study places emphasis on the low frequency model.

For unstable systems, the program described herein is still valid, however, the original higher-order system must first be modified by shifting the imaginary axis prior to the approximation. Thus for an unstable transfer function, $H(s)$, the equation must be changed to $H(s+a)$, where a is chosen sufficiently large so that $H(s)$ is asymptotically stable. This procedure is described in detail in [Ref 1], pp 332.

III. COMPUTER PROGRAM CRITERIA

A. GENERAL

The program, referred to as ROUTH1, was written with the following criteria:

1. Minimum utilization of computer time

ROUTH1 consists of less than 150K of storage and takes less than 2 minutes of computer time. This not only conserves efficiency, but also provides the user with the desired data at a minimum cost.

2. Ease of use

Input data required consists of eleven data cards for maximum utilization of the program. Emphasis is placed on the ease of using the program to obtain the desired results with minimum time expended on computer programming.

3. Usefulness of output

a. The first, second, third and fourth order approximants to the original equation are printed in transfer function format. Both numerator and denominator coefficients are printed in ascending powers of S .

b. The roots of the original and reduced equations are provided to enable the user to study the response of the systems in the frequency domain.

c. Choice of variables for print out in table form is available with up to eight variables maximum for any one of three possible runs.

d. Choice of variables for graphical output with up to four curves may be plotted separately or all on one graph.

e. Graphical output response to input excitations consisting of a Step, Ramp, or Sinusoidal input are available. The graphs display the original output response compared to the lower order response and the error is plotted to display the differences in response. Multiple inputs may be used.

B. SUBROUTINES

Two subroutines are utilized in ROUTH1. The roots of the original and reduced equations are determined by subroutine PRQD, which was taken from [Ref 3], and the tables and graphs are determined by subroutine REDUCT1 which is a modification of INTEG1, [Ref 3].

IV. FOURTH ORDER EXAMPLE

A. THREE STEP PROCEDURE

To illustrate the "Routh Approximation Method", A fourth order example was chosen for simplicity.

The method, described in detail in [Ref 1], consists of three basic steps. The following transfer function is used to illustrate the procedure:

$$H(s) = \frac{1}{20 + 32s + 24s^2 + 8s^3 + s^4}$$

The first step is to compute what are termed Alpha and Beta coefficients from the Routh Table shown on page 19.

Alpha	20	24	1
	32	8	
Alpha ₁ =0.625	19	1	
Alpha ₂ =1.684	6.3158		
Alpha ₃ =3.008	1		
Alpha ₄ =6.3158			
<hr/>			
Beta	1	0	
	0	0	
Beta ₁ =0.03125	-0.25		
Beta ₂ =0.0	0.0		
Beta ₃ =0.03958			
Beta ₄ =0.0			

Step two in the procedure is to obtain what are termed the Routh convergents, which are based on the following:

Letting $A_k(s)$ and $B_k(s)$ denote the denominator and the numerator, respectively, of the k^{th} Routh convergent, i.e.,

$$A_1(s) = \text{Alpha}_1 s + 1$$

$$B_1(s) = \text{Beta}_1$$

$$A_2(s) = \text{Alpha}_1 \text{Alpha}_2 s^2 + \text{Alpha}_2 s + 1$$

$$B_2(s) = \text{Alpha}_2 \text{Beta}_1 s + \text{Beta}_2$$

⋮

The general expression from [Ref 1] is the following:

$$A_k(s) = \text{Alpha}_k(s) A_{k-1}(s) + A_{k-2}(s)$$

$$B_k(s) = \text{Alpha}_k s B_{k-1}(s) + B_{k-2}(s) + \text{Beta}_k \quad k=1,2,3\dots$$

$$\text{with } A_{-1}(s)=0.0 \quad B_{-1}(s)=0.0$$

$$A_0(s) = 1.0 \quad B_0(s) = 0.0$$

The Routh convergents for the fourth order example are the following:

$$R_1(s) = \frac{0.03125}{0.625s + 1}$$

$$R_2(s) = \frac{0.056316}{1.05263s^2 + 1.6842s + 1.0}$$

$$R_3(s) = \frac{0.158833s^2 - 0.0083}{3.166s^3 + 5.066s^2 + 3.633s + 1}$$

$$R_4(s) = \frac{s^3}{20s^4 + 32s^3 + 24s^2 + 8s + 1.0}$$

The third, and final step in the procedure is to apply what is termed a reciprocal transformation, defined by

$$H_k(s) = \frac{1}{s} \times R_k(1/s)$$

which is merely a reversal of the order of the polynomial coefficients. Thus, for the example given, the reduced order approximations are given by the following:

$$H_1(s) = \frac{0.03125}{s + 0.625}$$

$$H_2(s) = \frac{0.056316}{s^2 + 1.6842s + 1.05263}$$

$$H_3(s) = \frac{-0.0083s^2 + 0.158333}{s^3 + 3.633s^2 + 5.066s + 3.166}$$

$$H_4(s) = \frac{1}{s^4 + 8s^3 + 24s^2 + 32s + 20}$$

As expected, the fourth-order approximation is the same as the original equation.

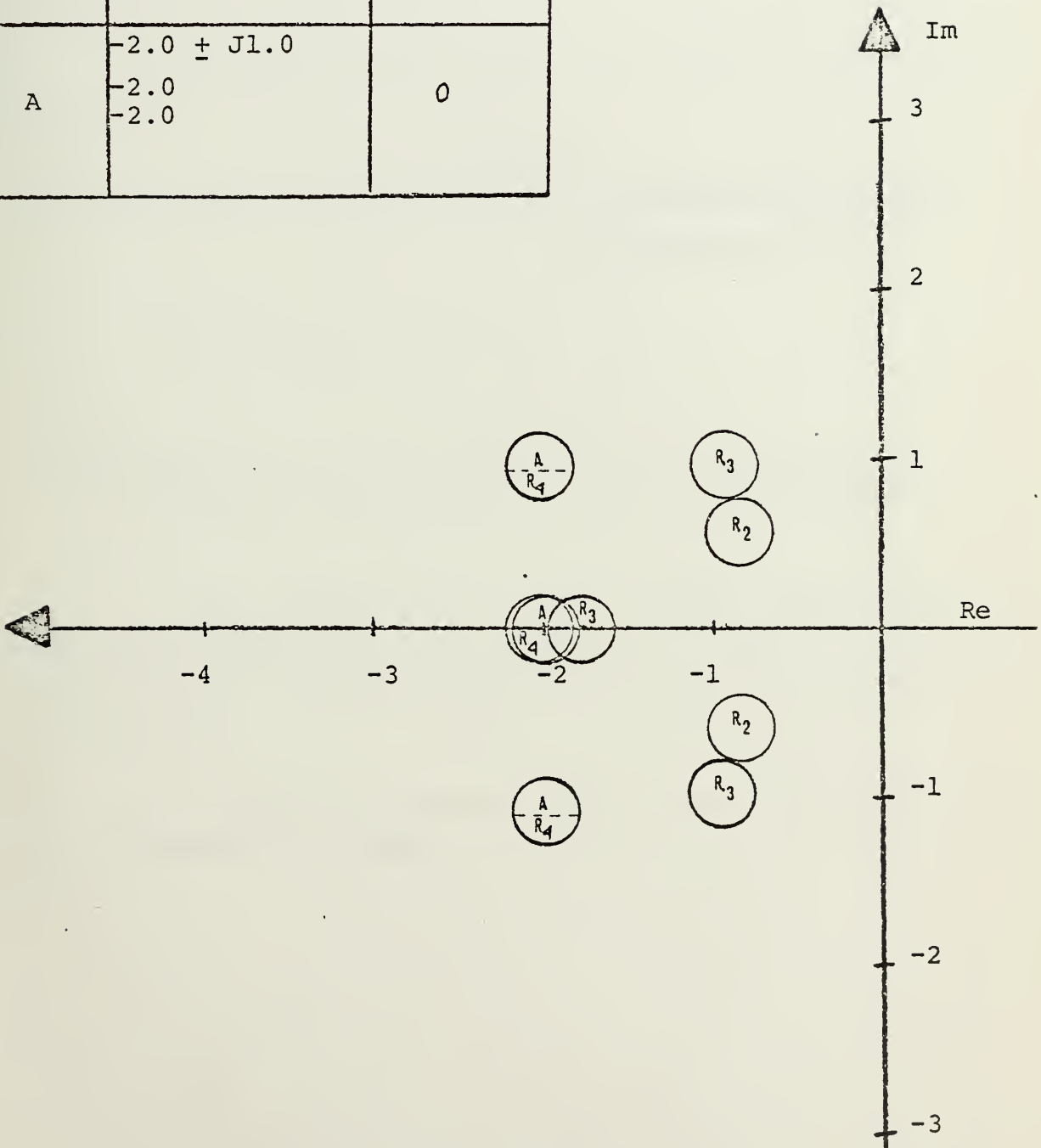
The poles of the approximants are illustrated in table IV.2. The poles of the approximants approach the dominant poles of the original higher-order equation as the order of the approximant is increased.

	C_{d1}	C_{d2}	C_{d3}	C_{d4}	...
	C_{d5}	C_{d6}	C_{d7}	...	
$\text{Alpha}_1 = C_{d1}/C_{d5}$	$W_1 = C_{d2} - \text{Alpha}_1 C_{d6}$	$W_2 = C_{d3} - \text{Alpha}_1 C_{d7}$	$W_3 = C_{d4} - \text{Alpha}_1 C_{d8}$...	
$\text{Alpha}_2 = C_{d5}/W_1$	$W_4 = C_{d6} - \text{Alpha}_2 W_2$	$W_5 = C_{d7} - \text{Alpha}_2 W_3$...		
$\text{Alpha}_3 = W_1/W_4$	$W_6 = W_2 - \text{Alpha}_3 W_5$	$W_7 = W_3 - \text{Alpha}_3$...		
$\text{Alpha}_4 = W_4/W_6$	$W_8 = W_5 - \text{Alpha}_4 W_7$...			
...					
	C_{n1}	C_{n2}	C_{n3}	C_{n4}	...
	C_{n5}	C_{n6}	C_{n7}	C_{n8}	...
$\text{Beta}_1 = C_{n1}/C_{d5}$	$U_1 = C_{n2} - \text{Beta}_1 C_{d5}$	$U_2 = C_{n3} - \text{Beta}_1 C_{d6}$...		
$\text{Beta}_2 = C_{n5}/W_1$	$U_3 = C_{n6} - \text{Beta}_2 W_2$	$U_4 = C_{n7} - \text{Beta}_2 W_3$...		
$\text{Beta}_3 = U_1/W_4$	$U_5 = U_2 - \text{Beta}_3 W_5$...			
$\text{Beta}_4 = U_3/W_6$	$U_6 = U_4 - \text{Beta}_4 W_7$...			
...					

ROUTH TABLE

Table IV.1

Poles and Zeros of Approximant		
Order	Poles	Zeros
R_2	$-0.842 \pm j0.586$	0
R_3	$-0.920 \pm j0.958$ -1.792	± 4.36
R_4	$-2.0 \pm j1.0$ -2.0 -2.0	0
A	$-2.0 \pm j1.0$ -2.0 -2.0	0



POLES OF ROUTH APPROXIMANTS
Table IV.2

V. COMPUTER EXAMPLE

A. SEVENTH ORDER SYSTEM

A seventh order system with the transfer function given by [Ref 2], as

$$G(s) = \frac{384 \times 10^7}{s^7 + 432s^6 + 62670s^5 + 3615900s^4 + 75114000s^3 + 553920000s^2 + \dots + 1443200000s + 384 \times 10^7}$$

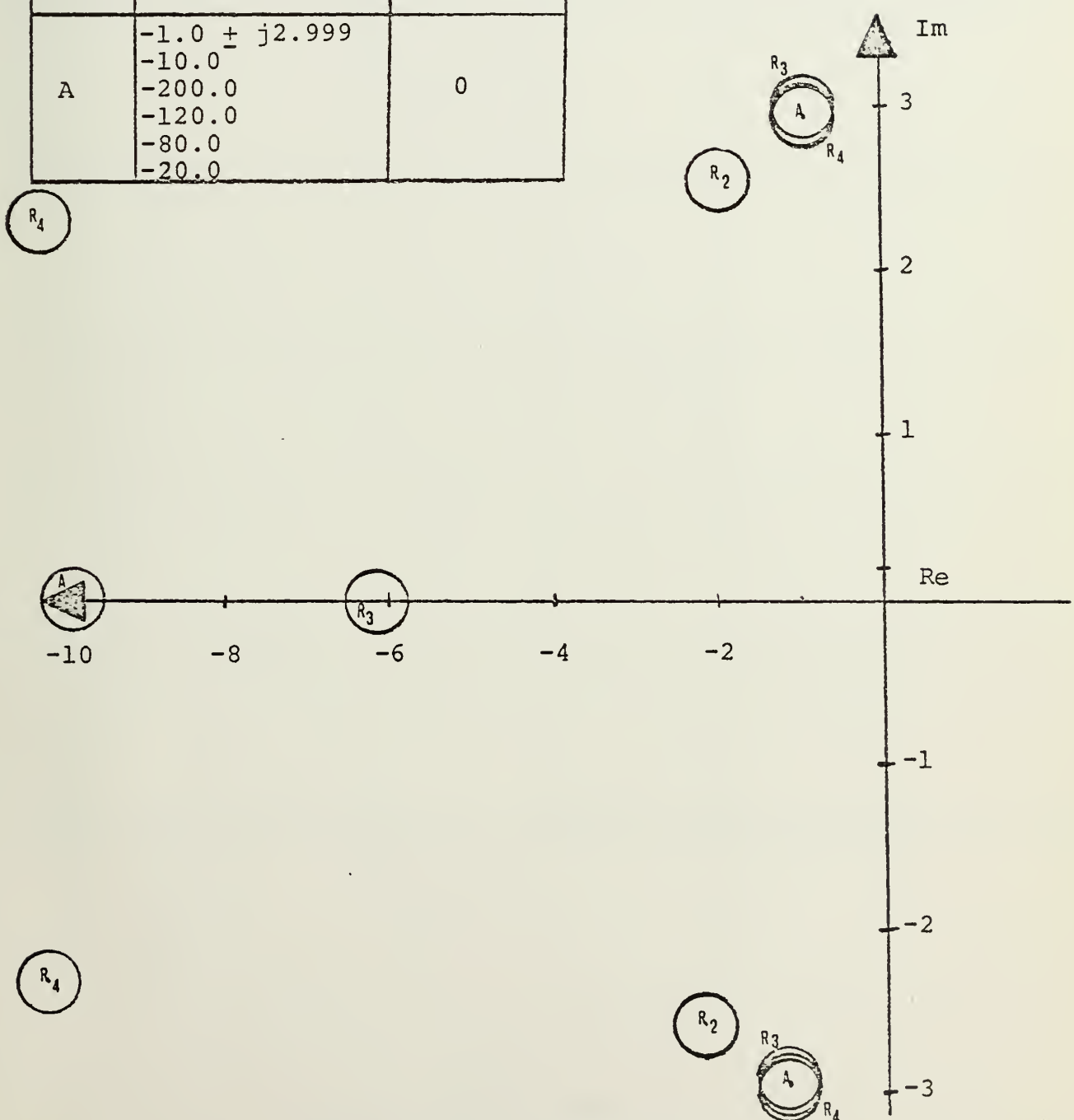
or in factored form as

$$G(s) = \frac{384 \times 10^7}{(s^2 + 2s + 10)(s + 10)(s + 20)(s + 80)(s + 120)(s + 200)}$$

was reduced to the low order approximants by utilizing ROUTH1 and the response to input excitations, consisting of a Step and Ramp are illustrated in figures 5.1 through 5.6.

The roots of the system and its low order approximants are illustrated in table V.1.

Poles and Zeros of Approximant		
Order	Poles	Zeros
R_2	$-2.038 \pm j2.586$	0
R_3	$-1.084 \pm j2.970$ -6.291	± 10.132
R_4	$-1.0 \pm j2.999$ $-10.89 \pm j2.30$	± 316.58
A	$-1.0 \pm j2.999$ -10.0 -200.0 -120.0 -80.0 -20.0	0



POLES OF ROUTH APPROXIMANTS
Table V.1

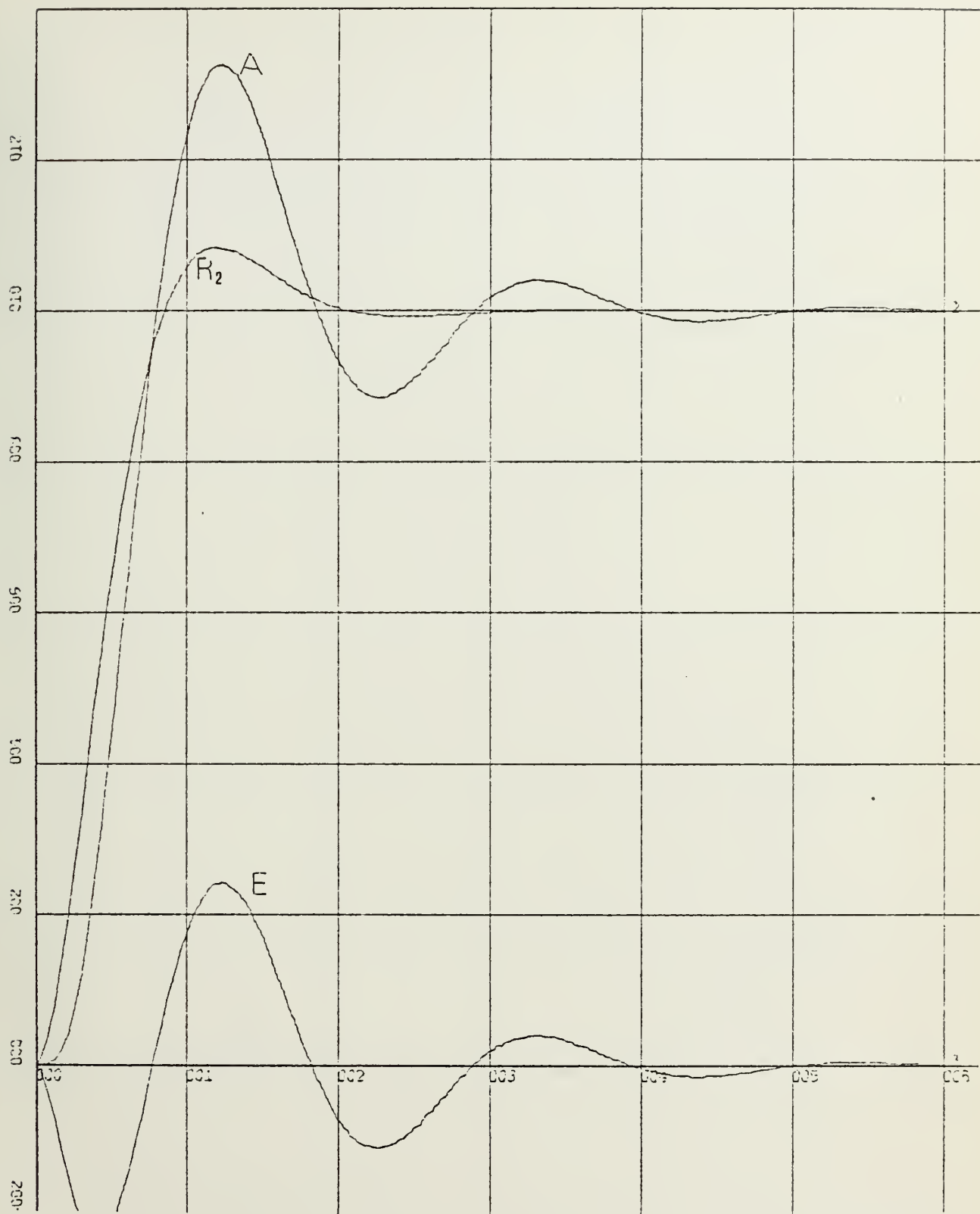


Figure 5.1

STEP INPUT

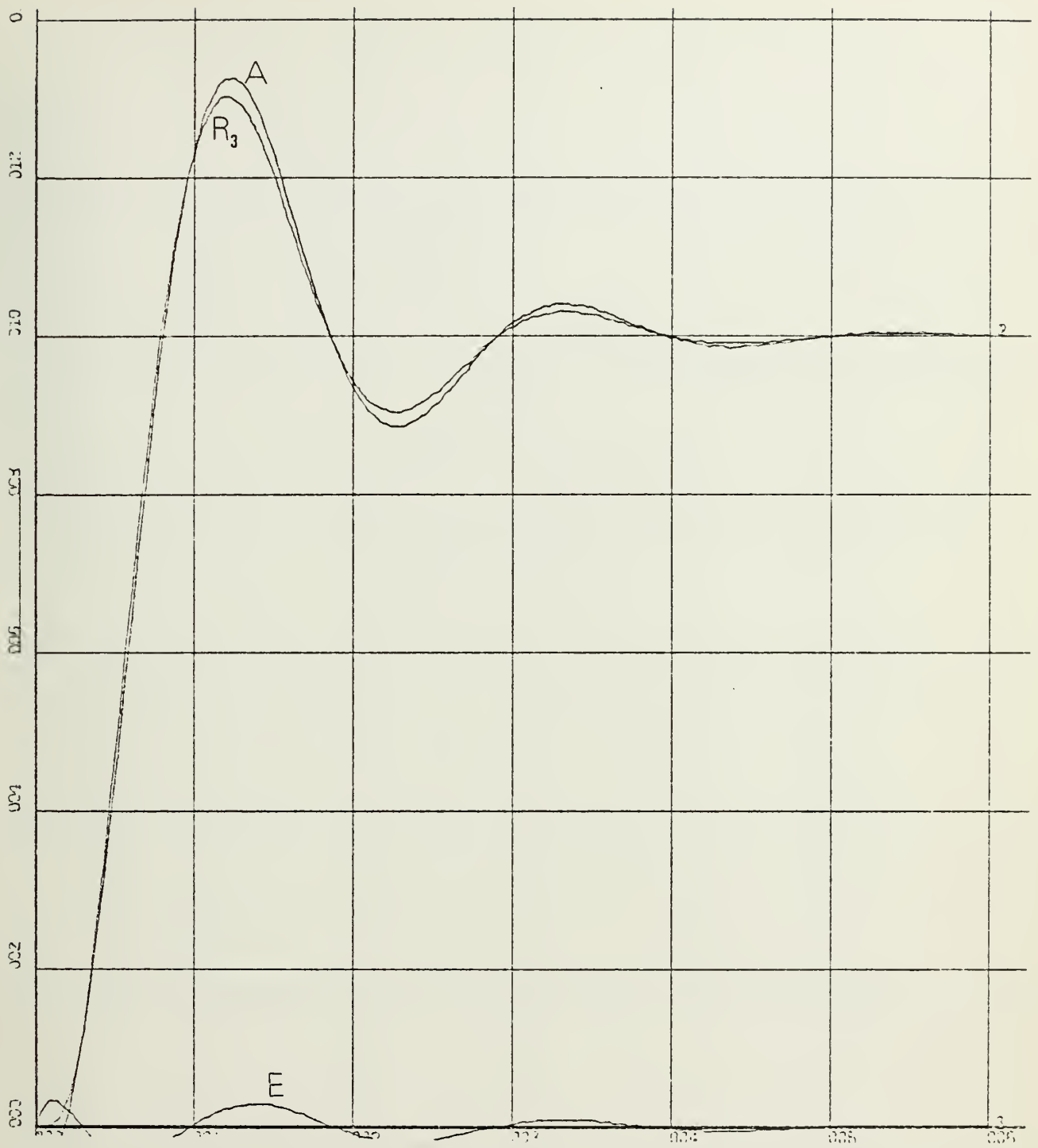


Figure 5.2
STEP INPUT

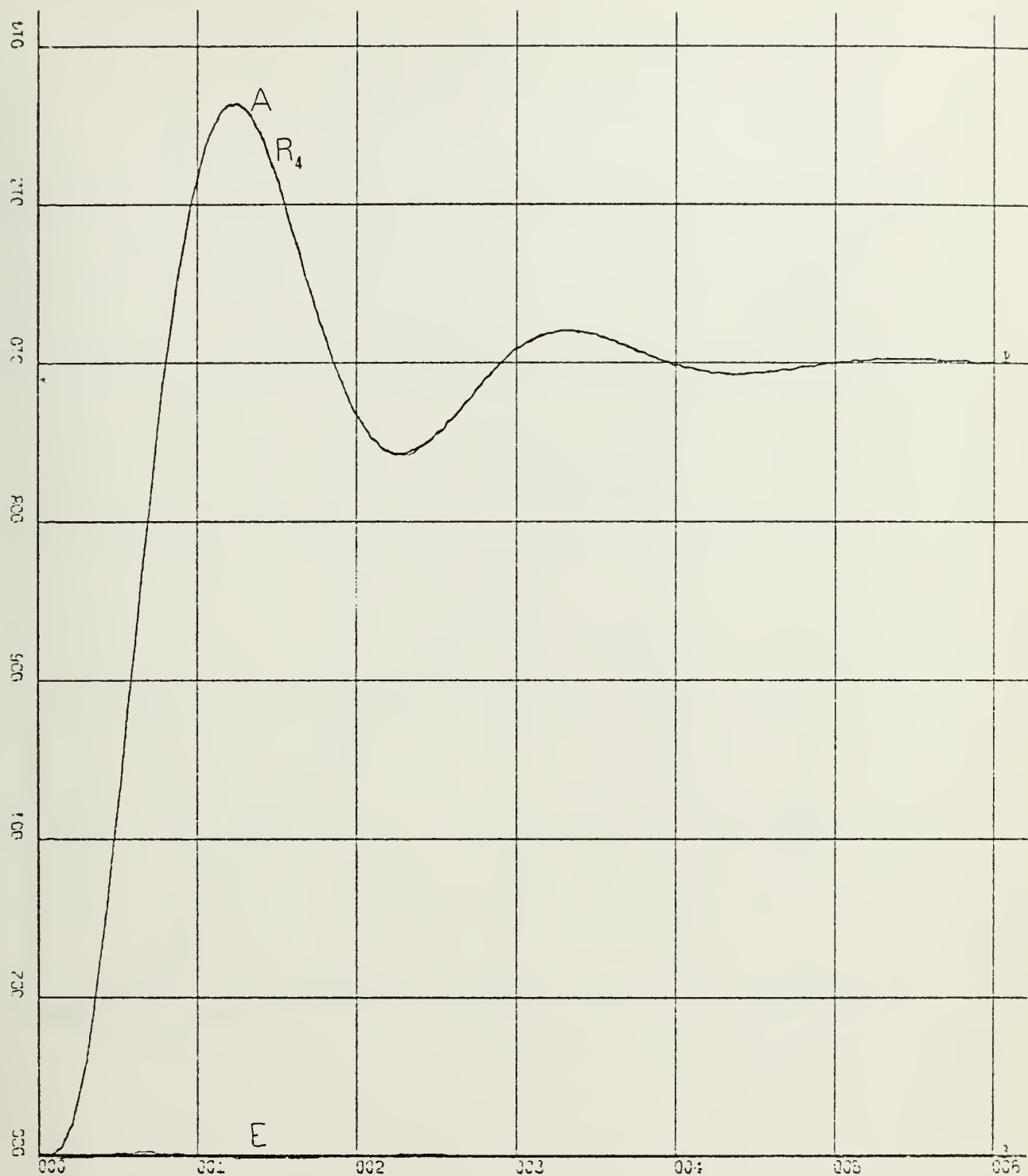


Figure 5.3
STEP INPUT

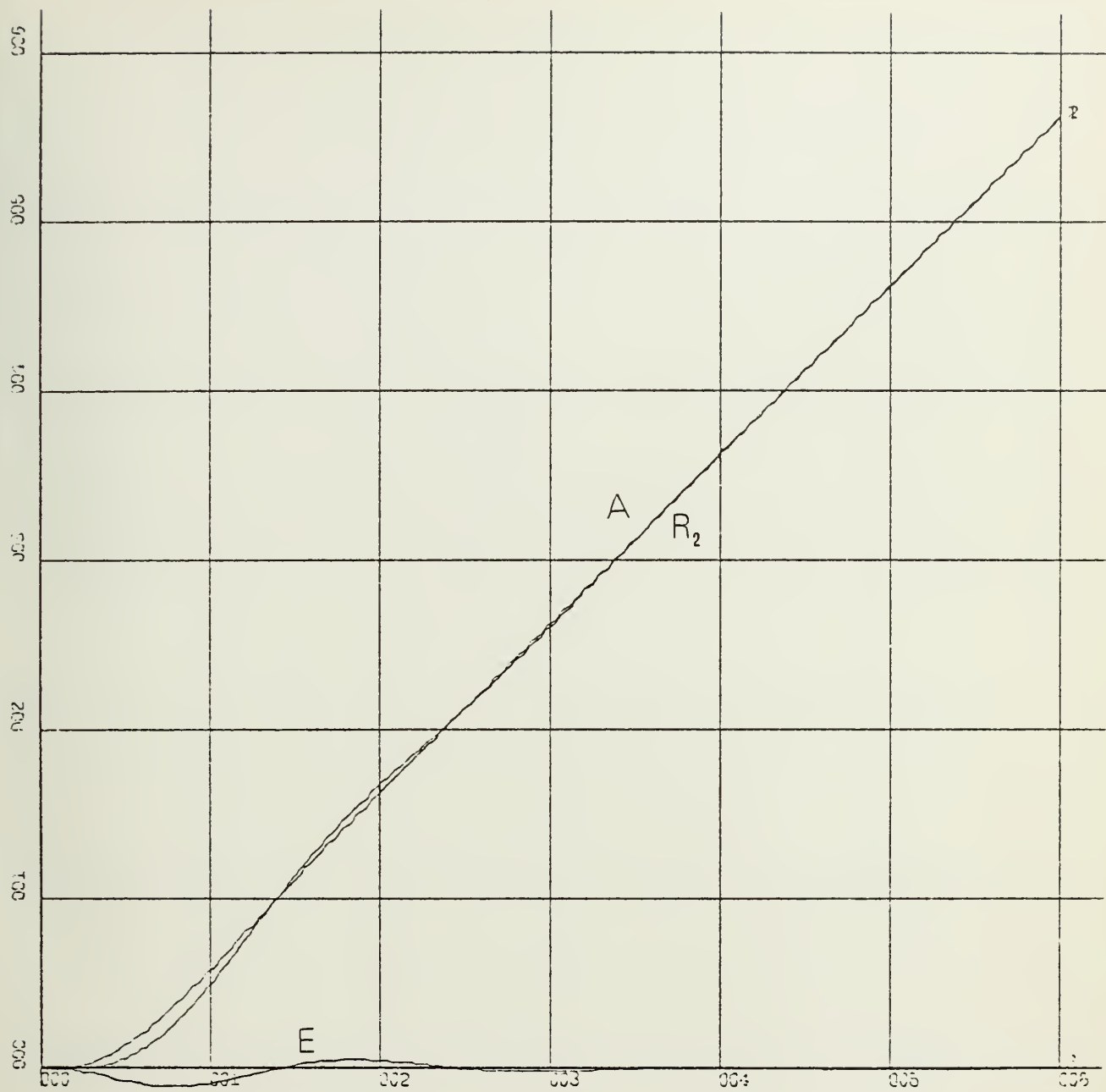


Figure 5.4

RAMP INPUT

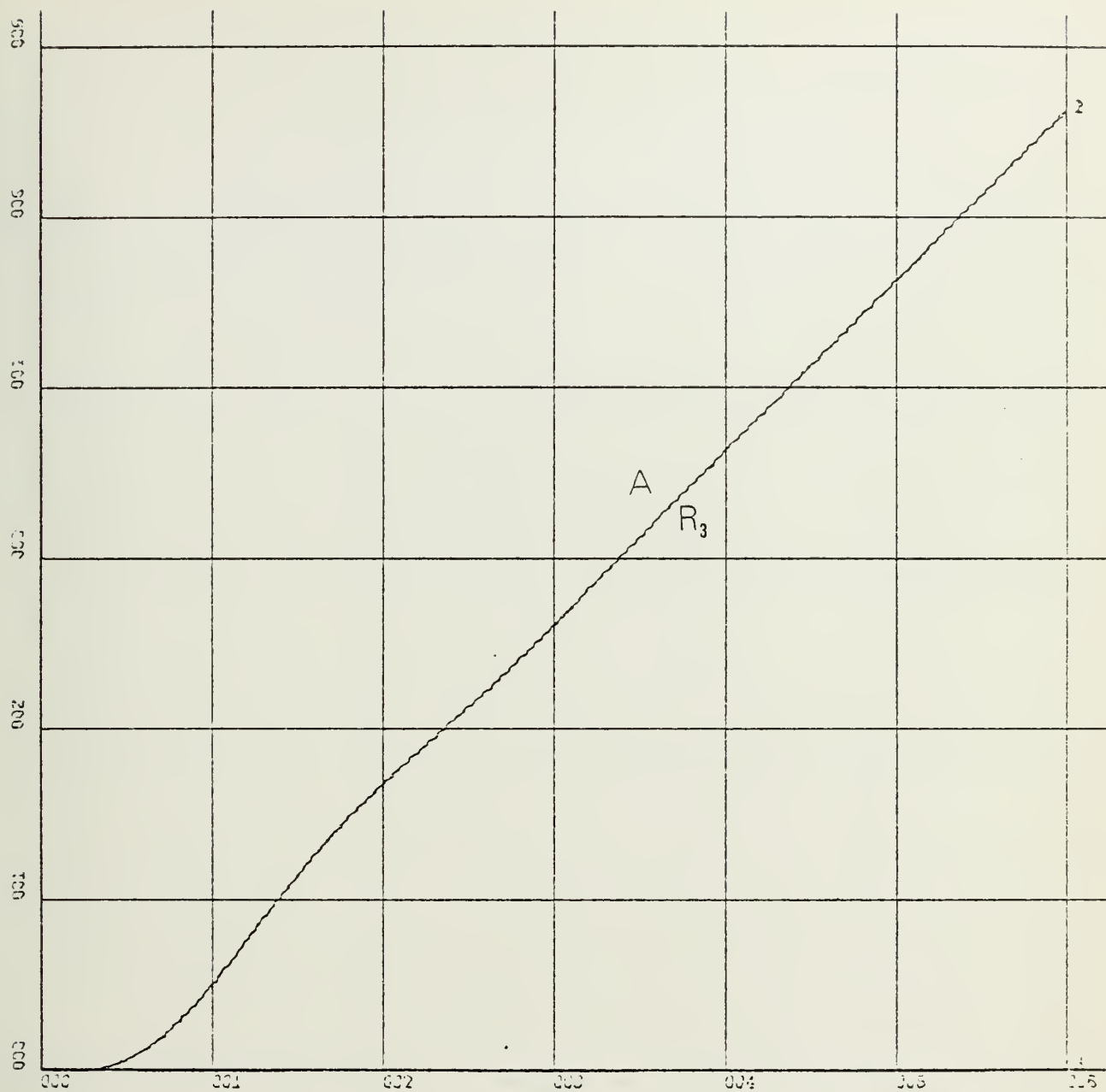


Figure 5.5

RAMP INPUT

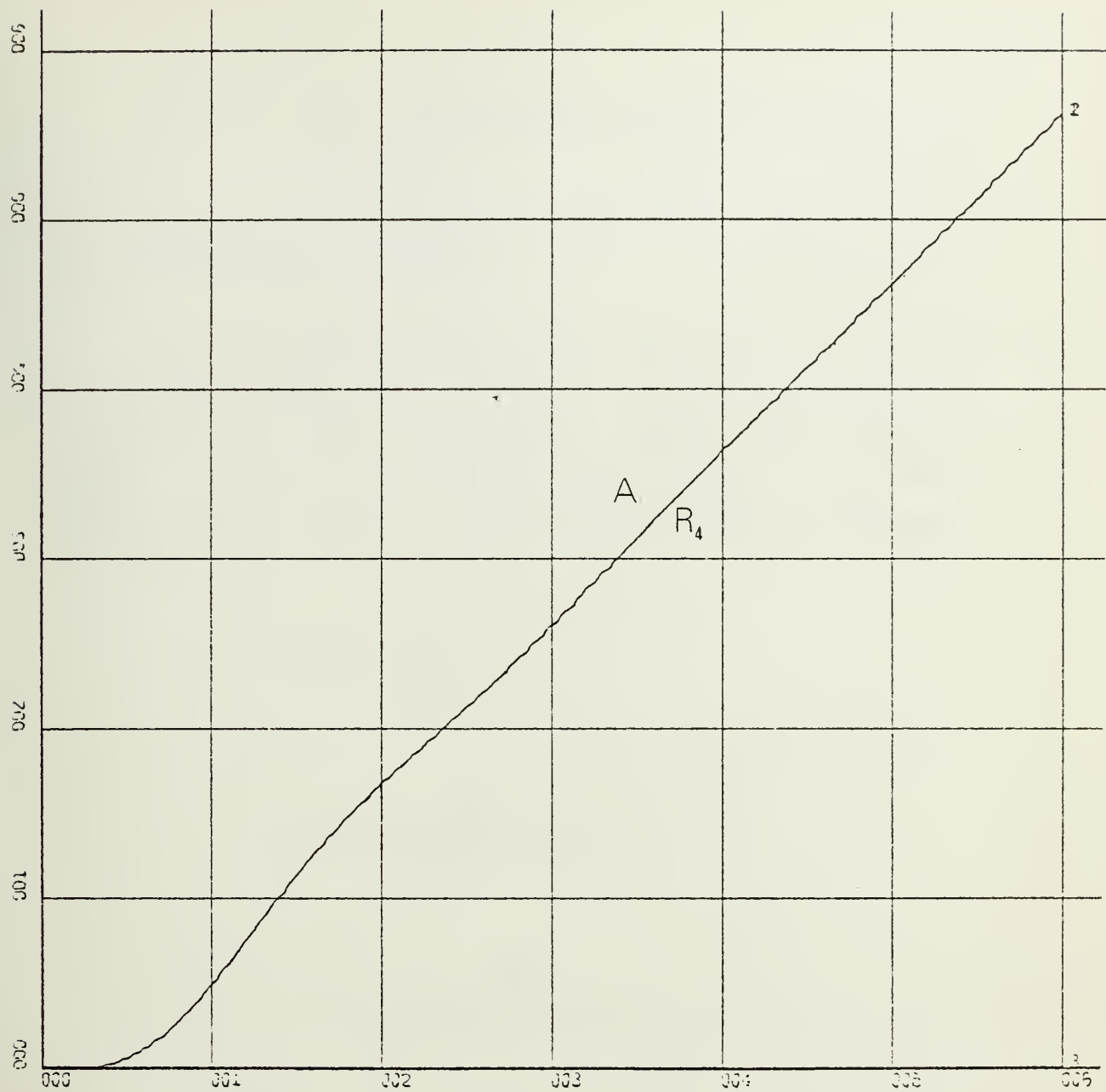


Figure 5.6

RAMP INPUT

VI. COMPARISON TO OTHER METHODS

A. DOMINANT POLE METHOD

The Dominant Pole Approximation method is based upon utilizing the poles closest to the imaginary axis. The equation must be factored to obtain the characteristic roots. For the given seventh-order system, the lower-order equations are given by this method as the following:

$$H(s) = \frac{10}{s^2 + 2s + 10}$$

$$H(s) = \frac{100}{s^3 + 14s^2 + 30s + 100}$$

$$H(s) = \frac{2000}{s^4 + 34s^3 + 310s^2 + 700s + 2000}$$

Graphical plots of the Dominant Pole reduced equations, in response to Step inputs, are illustrated in figures 6.1 through 6.3 in comparison to the original equation and figures 6.4 through 6.6 illustrate the comparison to the Routh equations. The error between the systems is also plotted. Table VI.1 gives the analytical data for performance measure comparisons.

B. ITERATIVE OPTIMIZATION METHOD

The following equations, representing the seventh-order example, were taken from [Ref 2]. This method makes use of an iterative minimization technique to locate the best pole and zero locations for the lower-order models.

$$H(s) = \frac{7.203856}{s^2 + 1.98616s + 7.203856}$$

$$H(s) = \frac{52.2861}{s^3 + 7.1383s^2 + 19.5015s + 52.2861}$$

$$H(s) = \frac{1470.1403}{s^4 + 28.5204s^3 + 209.3842s^2 + 552.5241s + 1470.1403}$$

Plots of these equations versus the Original equation and the Routh equations are illustrated in figures 6.7 through 6.12 in response to Step inputs. Table VI.1 provides numerical data for performance measure comparisons.

C. PRESENTATION OF DATA

Comparisons were made between the "Routh Approximation Method", and that of the Dominant Pole and Iterative Optimization methods previously described.

The basis for comparison consists of the following criteria:

1. Peak Overshoot--- $M_{pt} = Y_{\max}/Y_{ss}$
2. Delay Time----- T_d =time for $Y(t)$ to reach 0.5 Y_{ss} the very first time.
3. Rise Time----- T_r =time for $Y(t)$ to go from 0.1 to 0.9 of the final value. $T_r = 1/BW$.
4. Settling Time---- T_s =time at which $Y(t) = Y_{ss}$
5. Graphical Representation in response to Step inputs.
6. Average Error---- $J = \left| \sum E/t_i \right|$, t_i =integration steps

Table VI.1 illustrates the above comparisons.

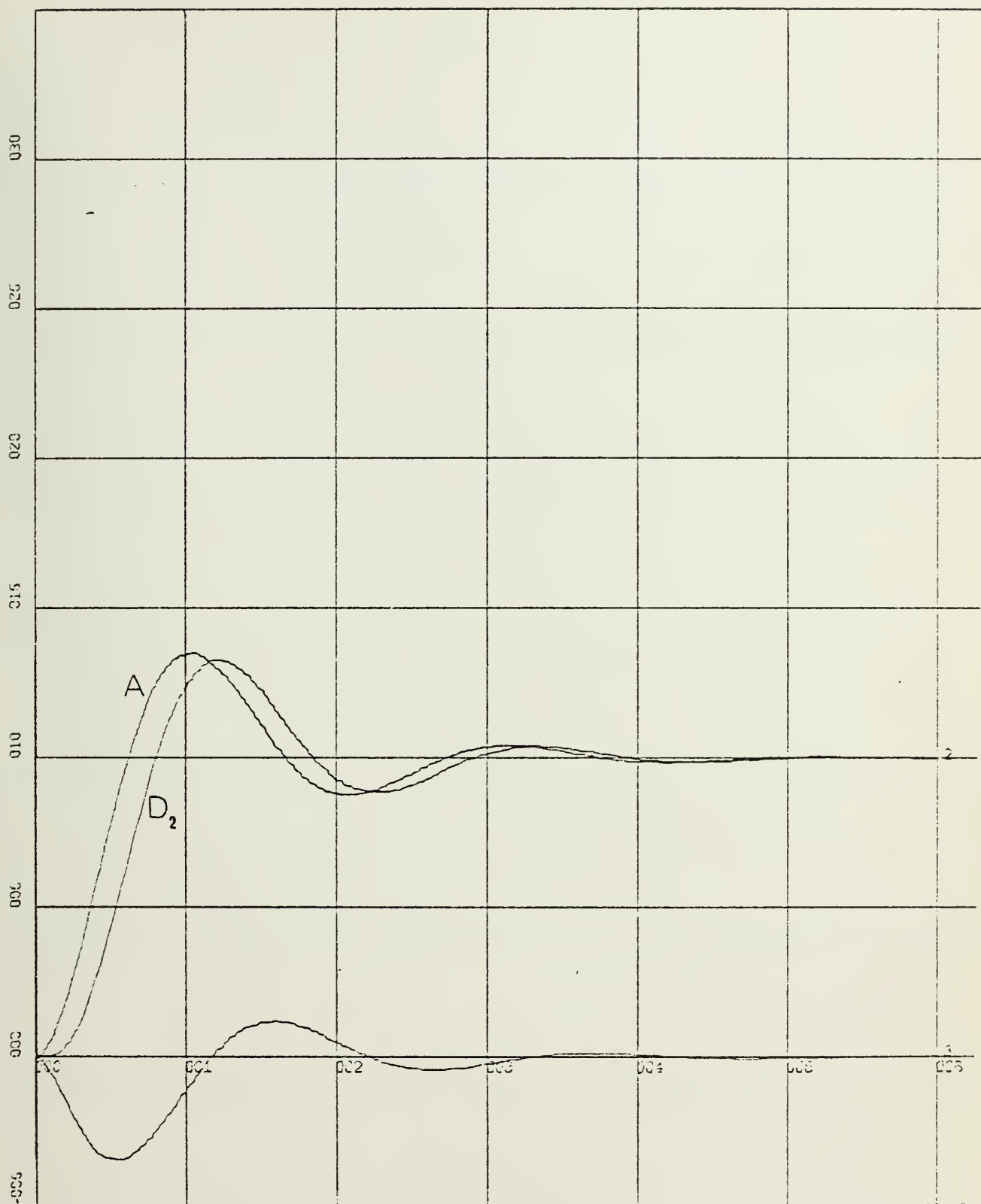


Figure 6.1
STEP INPUT

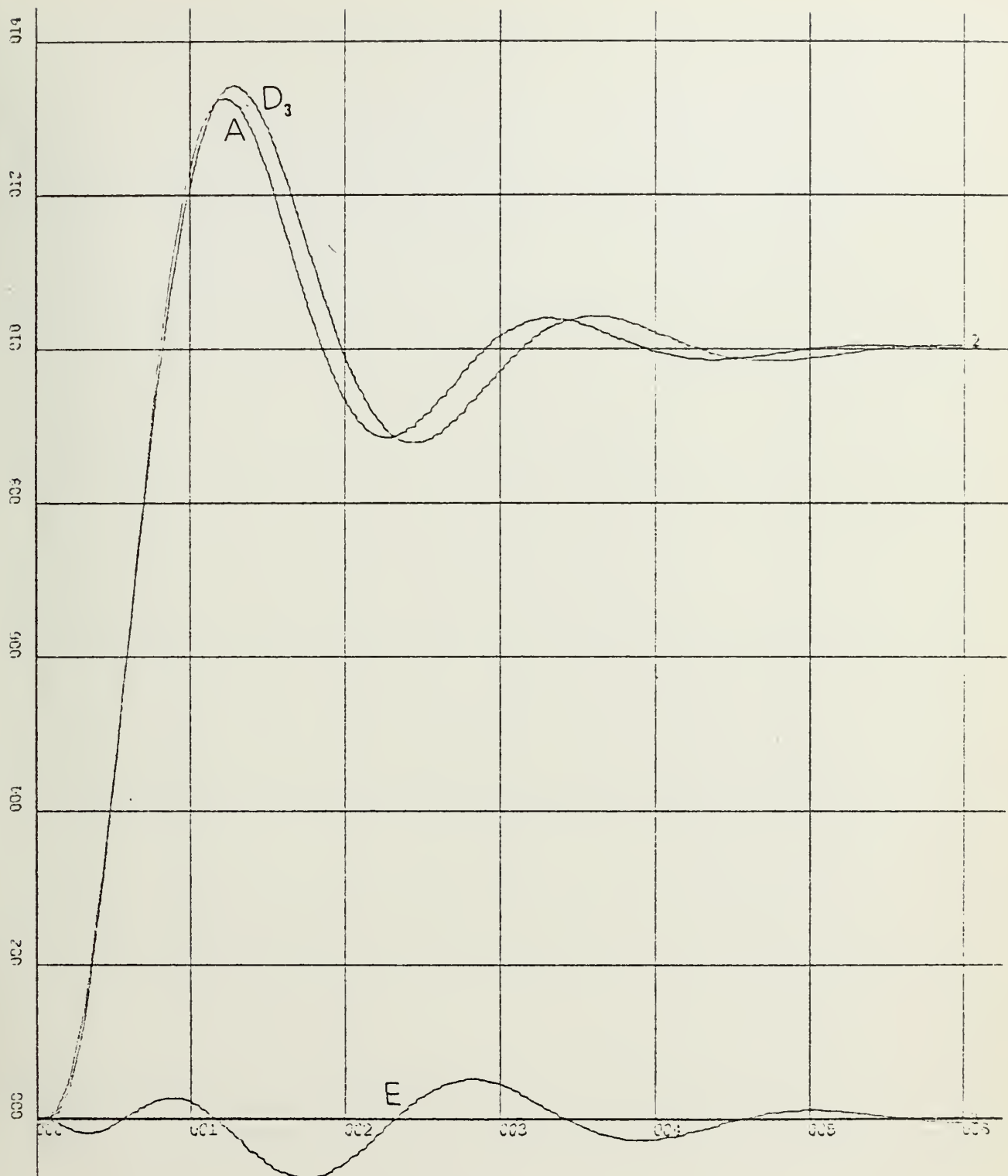


Figure 6.2
STEP INPUT

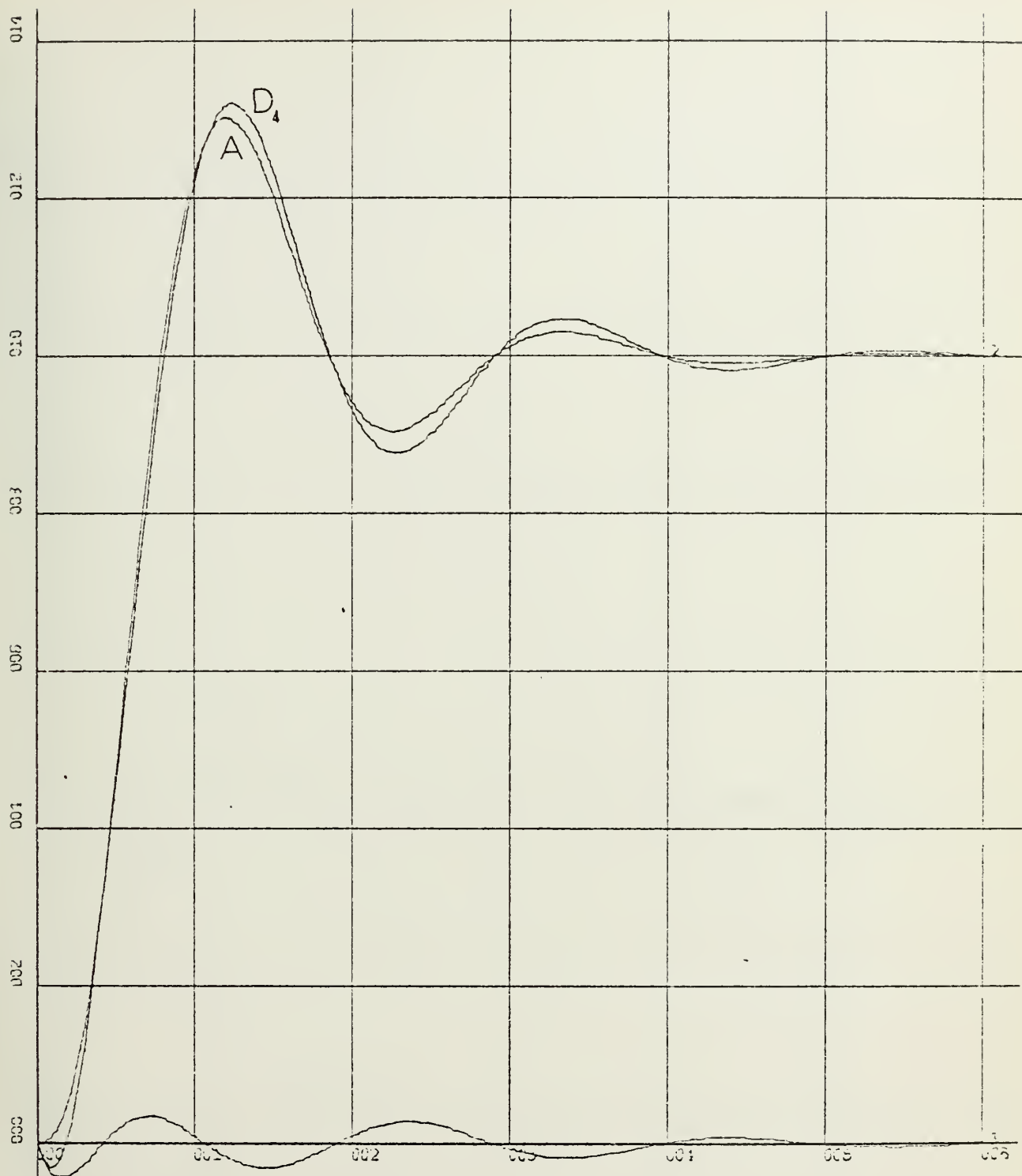


Figure 6 .3

STEP INPUT

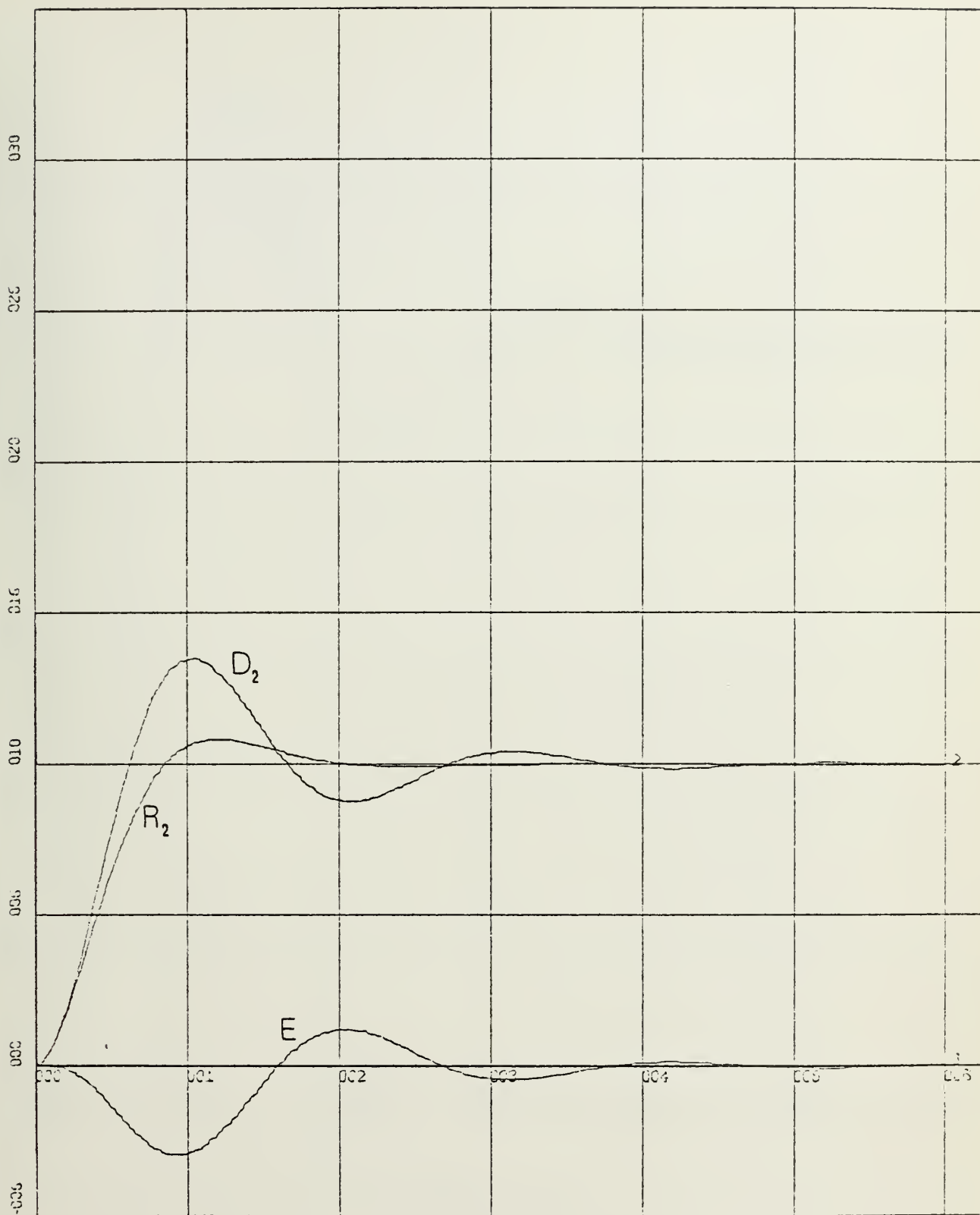


Figure 6 .4

STEP INPUT

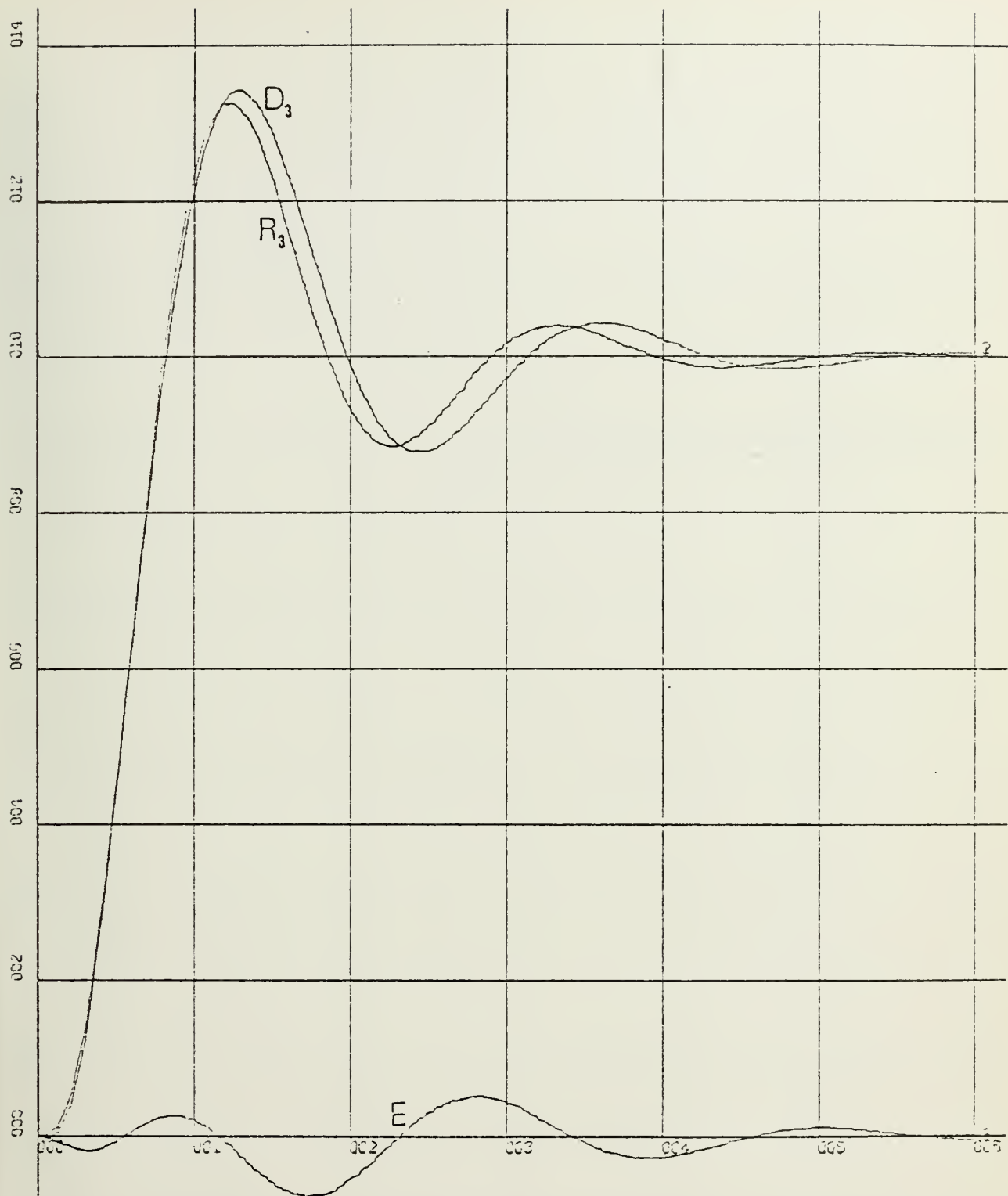


Figure 6 .5

STEP INPUT

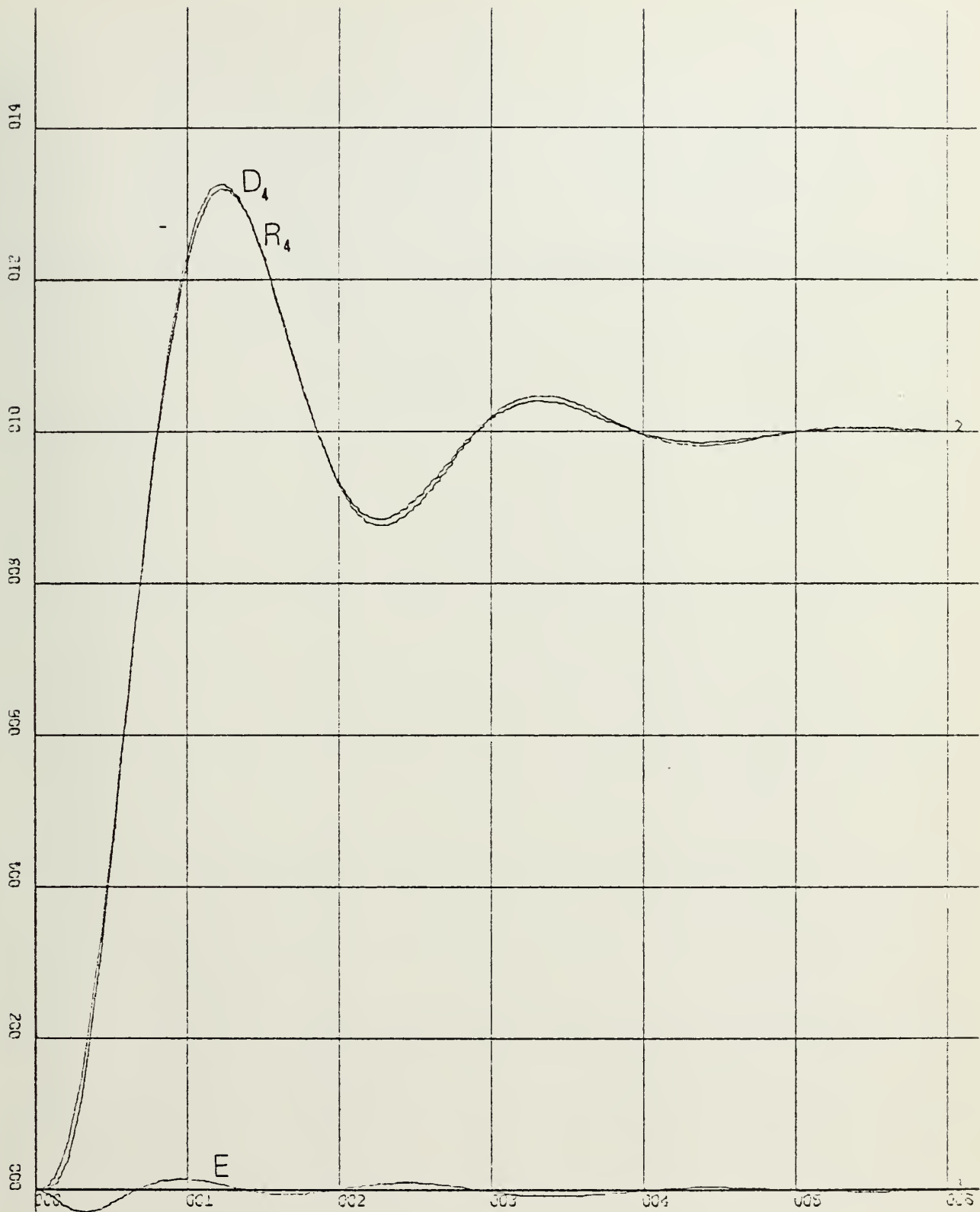


Figure 6.6
STEP INPUT

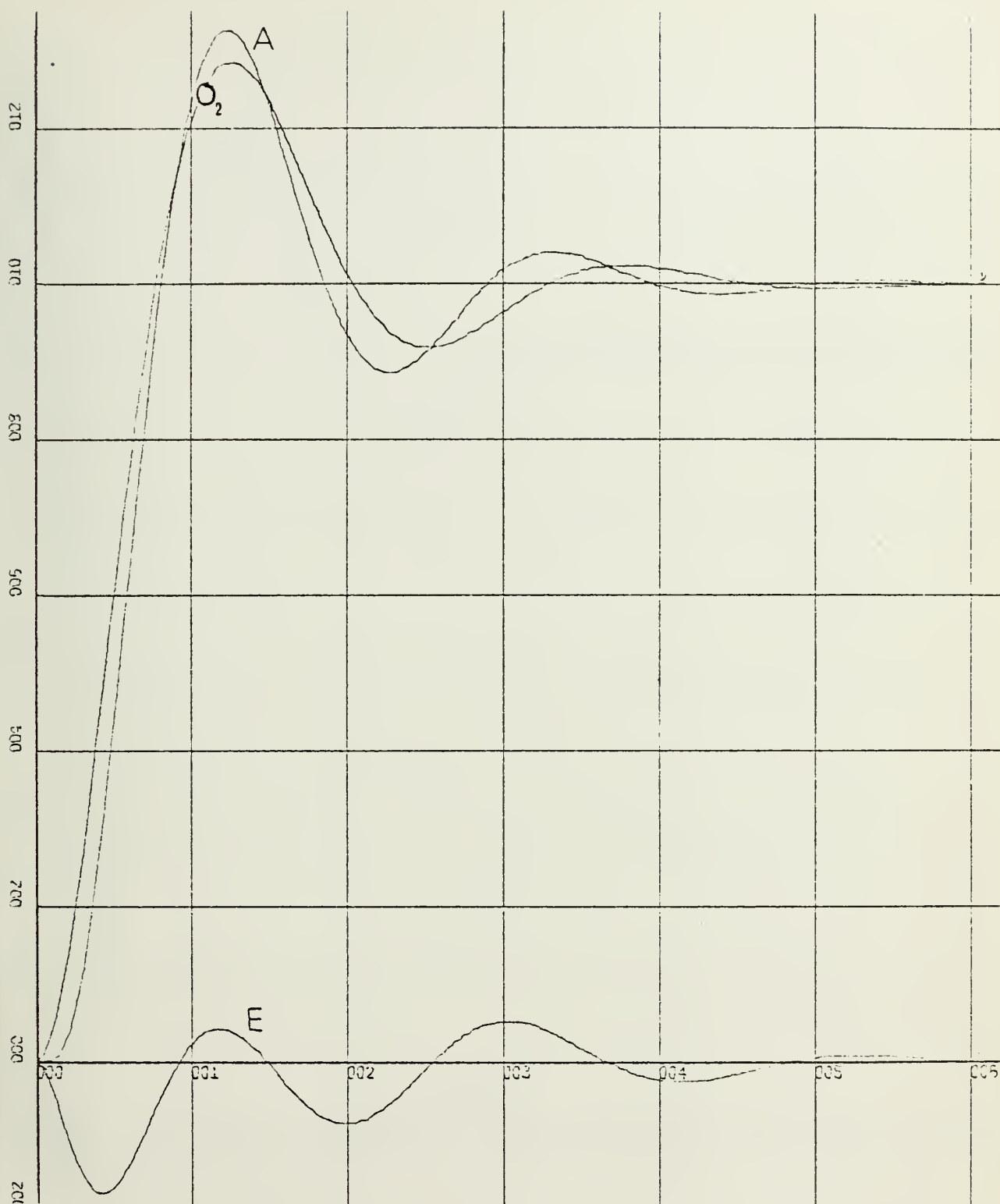


Figure 6.7
STEP INPUT

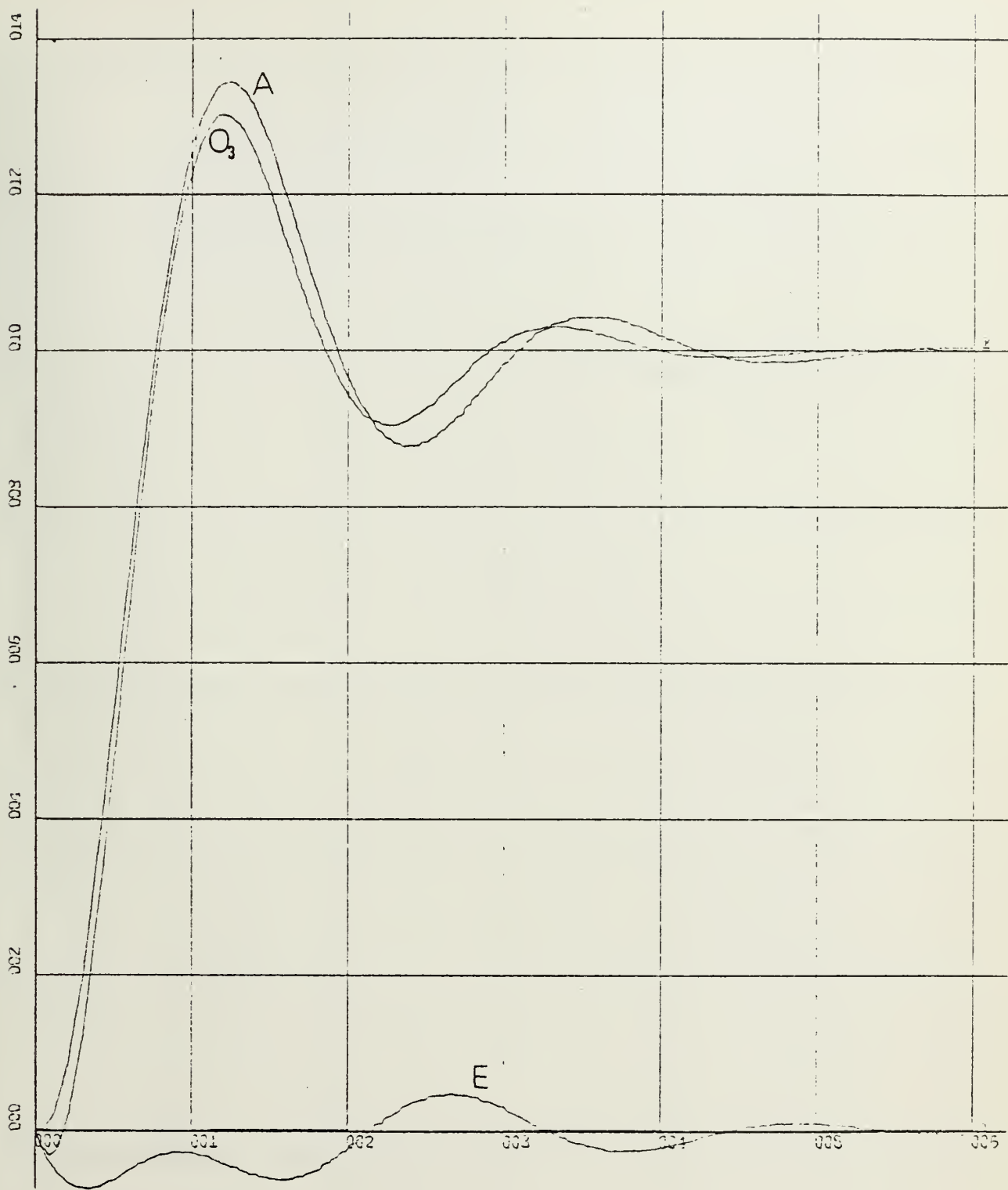


Figure 6.8
STEP INPUT



Figure 6.9
STEP INPUT

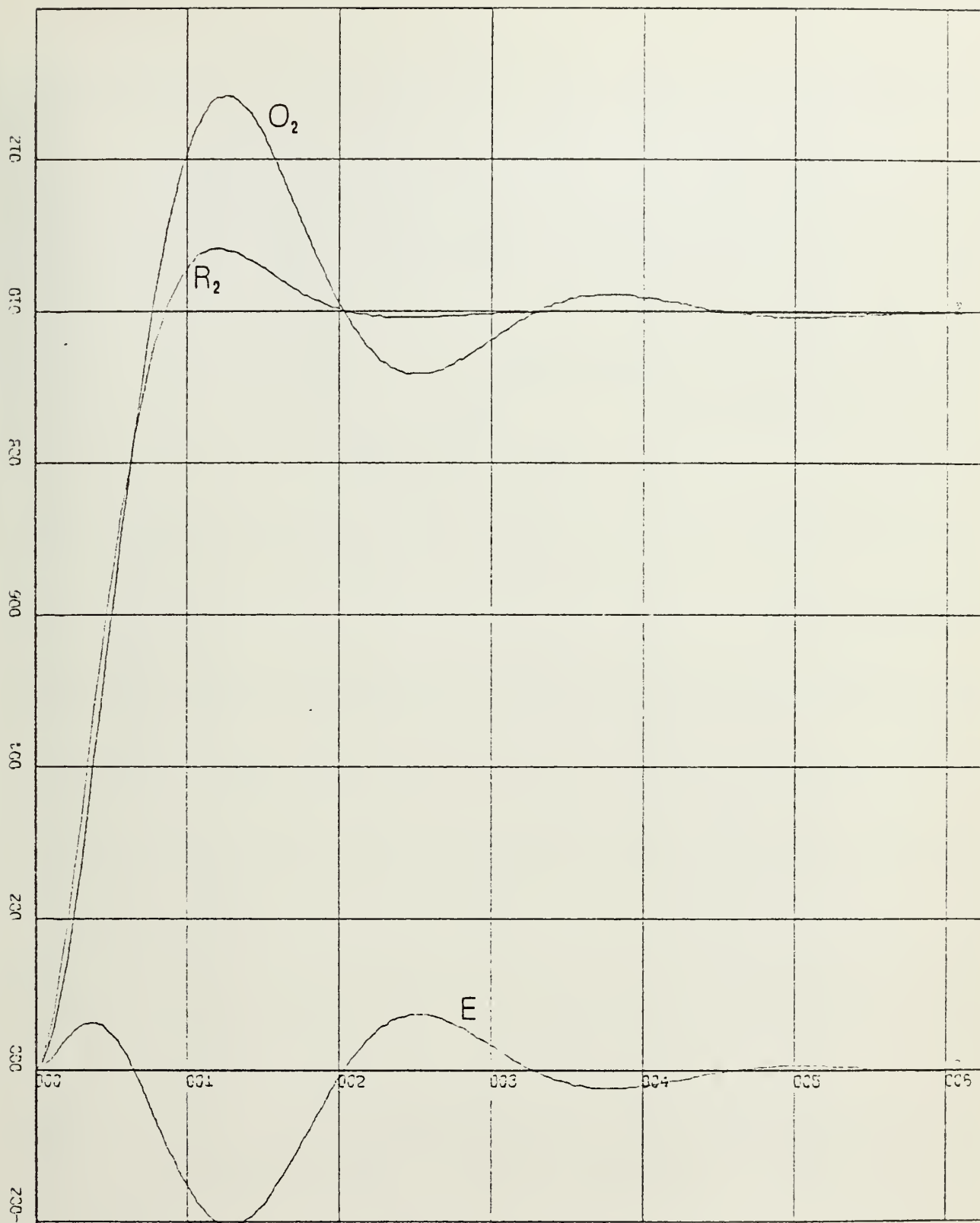


Figure 6 10
STEP INPUT

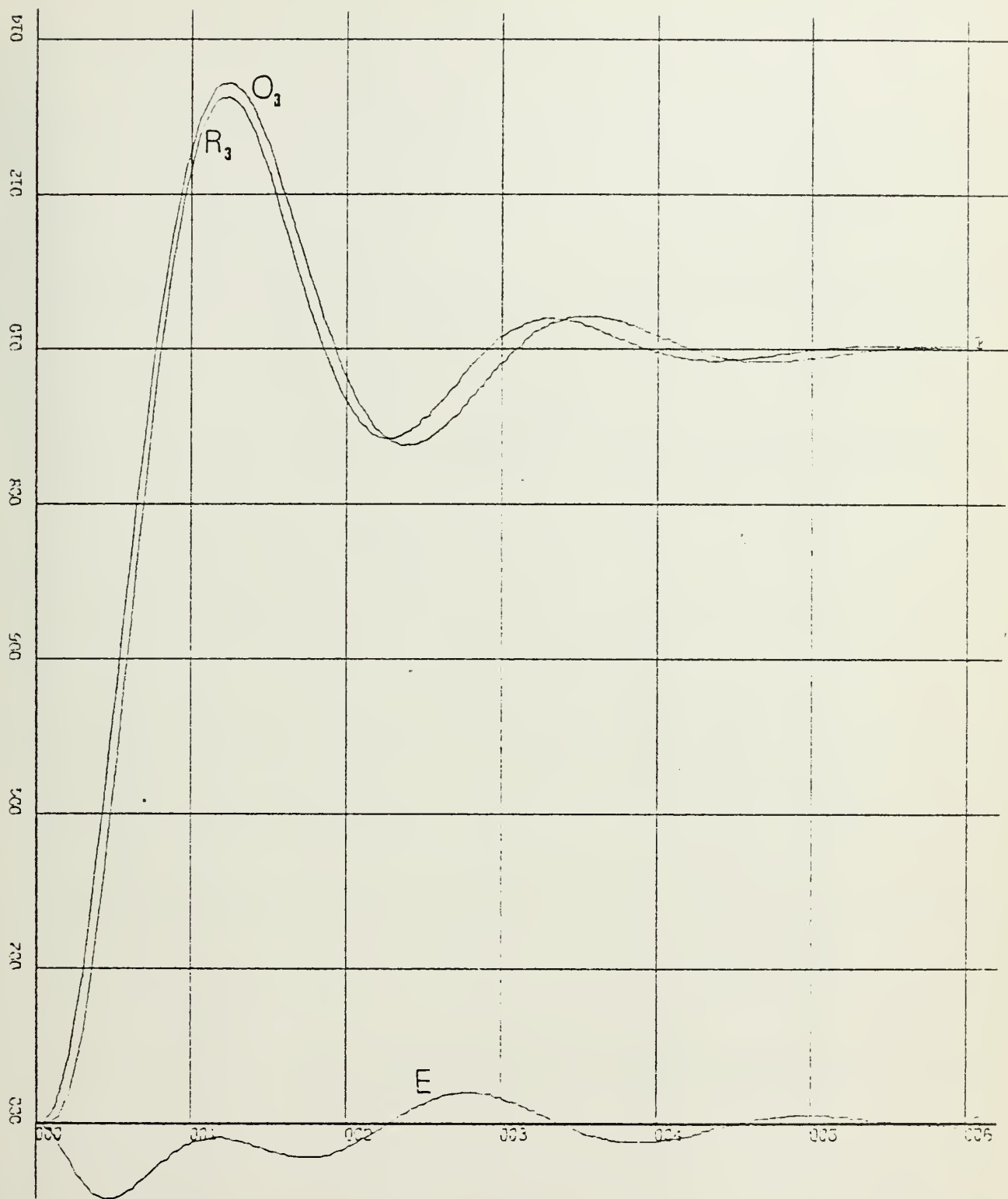


Figure 6.11

STEP INPUT

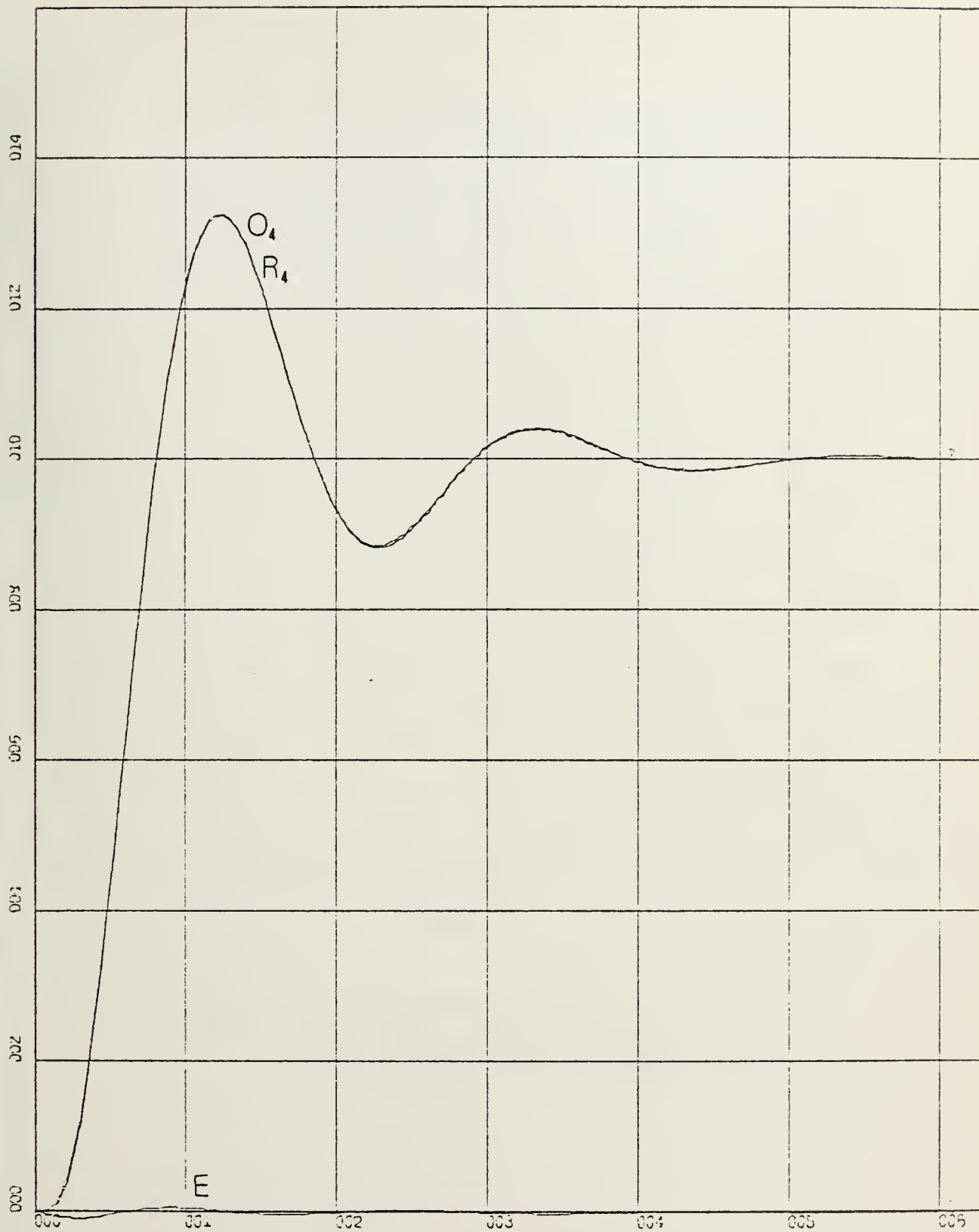


Figure 6.12
STEP INPUT

SYSTEM TYPE	Mpt	Td	Tr	Ts	J
A	1.327	0.533	0.711	5.866	-----
R-4	1.329	0.533	0.711	5.866	3×10^{-6}
O-4	1.325	0.533	0.711	5.866	6×10^{-6}
D-4	1.338	0.528	0.711	5.866	1.09×10^{-4}
R-3	1.247	0.533	0.711	5.866	0.010
O-3	1.320	0.533	0.711	5.866	0.006
D-3	1.346	0.502	0.711	5.866	1.15×10^{-4}
R-2	1.083	0.400	0.711	5.866	0.062
O-2	1.285	0.533	0.711	5.866	0.037
D-2	1.350	0.412	0.684	5.866	0.059

TABLE VI.1

Table
Symbols A = Original Seventh Order
 R = Routh Approximation
 D = Dominant Pole Approximation
 O = Optimum Minimization Method

PERFORMANCE MEASURE COMPARISONS

VII. CONCLUSION

The graphical data and tabulated analysis of this thesis indicate that the Routh Approximation Method is a valuable formulation technique which produces very satisfactory results in acquiring approximations to higher-order systems with minimum cost.

In comparison to the other methods of analysis, the Routh Approximation Method offers a quick and easy analytical approach to obtaining low order models. Unlike the Pad'e Approximation, the Routh method ensures stability of the lower-order models, if the original higher-order system is stable. The original system need not be factored, as in the Dominant Pole Approximation method.

The computer program, ROUTH1, utilized to acquire the reduced order equations, the roots of the equations and the graphical plots and numerical tables, takes considerably less time than the minimization technique utilized in reference 2. However, the minimization technique operates efficiently without prior knowledge of the system's transfer function. In the Routh method the availability of the transfer function is a necessary requirement.

In comparing the low order equations of the various methods discussed, the Routh method has proven to be a valid and efficient solution to the problem of obtaining good low order approximants to complex higher-order systems.

REDUCED EQUATIONS IN ASCENDING POWERS OF S

ORDER (1)

NUMER 0.266075E 01

DENCM 0.266075E 01 0.100000E 01

ORDER (2)

NUMER 0.108456E 02 0.0

DENCM 0.108456E 02 0.407614E 01 0.100000E 01

ORDER (3)

NUMER 0.628943E 02 0.0 -0.612693E 00

DENCM 0.628943E 02 0.236378E 02 0.245931E 01 0.100000E 01

ORDER (4)

NUMER 0.124036E 04 0.0 -0.123761E 01 0.0

DENCM 0.124036E 04 0.486156E 03 0.177684E 03 0.237975E 02 0.100000E 01

ROOTS OF DENOMINATOR OF ORDER 2 ERROR, 0

REAL PART	IMAG. PART
-0.200000E 03	0.0
-0.120000E 03	0.0
-0.799996E 02	0.0
-0.199999E 02	0.0
-0.599991E 01	0.0
-0.100001E 01	0.299993E 01
-0.100001E 01	-0.299993E 01

ROOTS OF DENOMINATOR OF ORDER 2 ERROR, 0

REAL PART	IMAG. PART
-0.203807E 01	0.258687E 01
-0.203807E 01	-0.258687E 01

ROOTS OF DENOMINATOR OF ORDER 3 ERROR, 0

REAL PART	IMAG. PART
-0.629165E 01	0.0
-0.108407E 01	0.297011E 01
-0.108407E 01	-0.297011E 01

ROOTS OF DENOMINATOR OF ORDER 4 ERROR, 0

REAL PART	IMAG. PART
-0.108982E 02	0.230245E 01
-0.108982E 02	-0.230245E 01
-0.100049E 01	0.299932E 01
-0.100049E 01	-0.299932E 01

ERROR CODES

IER=0 NO ERRORS
 IER=1 NO CONVERGENCE WITH FEASIBLE TOLERANCE
 IER=2 POLY IS DEGENERATE (CONSTANT OR ZERO)
 IER=3 SUBROUTINE ABANDONED (ZERO DIVISORS)
 IER=4 NO S-FRACTION EXISTS
 IER=-1 POOR ACCURACY IN CALCULATIONS

COMPUTER OUTPUT

THESIS-SEVENTH ORDER VS-REDUCED ORDER EQUATIONS
3 RUNS ARE CALLED FOR

INPUT DATA RECORD FOR RUN NUMBER 1

ORDER OF EQUATIONS = 19
INITIAL TIME = 0.0
FINAL TIME = 0.8000E 01
STEP SIZE = 0.8889E-02

THE ONLY NON-ZERO CONSTANT IS
C(1) = 0.1000E 01

ALL THE INITIAL CONDITIONS ARE ZERO

THE COLUMN HEADINGS AND THE CORRESPONDING VARIABLES ARE

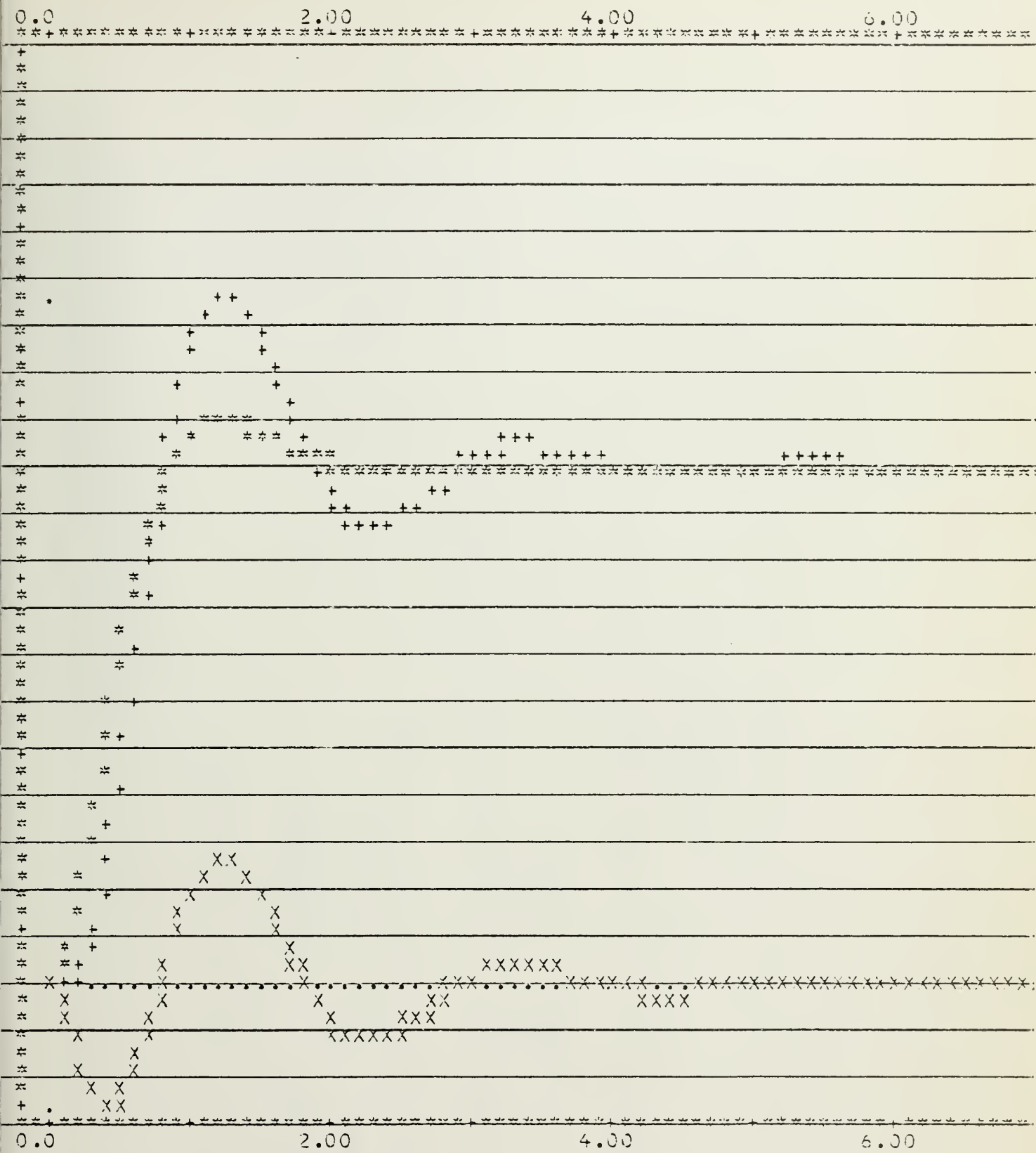
TIME	X(0)
DRIG	X(20)
SECOND	X(21)
ERROR	X(24)

THE GRAPH TITLE AND THE CORRESPONDING VARIABLES ARE

PESIT VS TIME	X(20) VS: X(0)
	X(21) VS: X(0)
	X(24) VS: X(0)

TESTS SEVENTH ORDER VS REDUCED ORDER EQUATIONS

TIME	CRIS	SECOND	ERROR
0.0	0.0	0.0	0.0
0.177778E CC	0.190333E -01	0.13236E 00	-0.11376E 00
0.355555E CC	0.17644E 00	0.40275E 00	-0.22631E 00
0.533333E CC	0.47702E 00	0.67507E 00	-0.19805E 00
0.711110E CC	0.81911E 00	0.38402E 00	-0.64906E -01
0.88888E 00	0.11050E 01	0.10127E 01	0.92278E -01
0.10666E 01	0.12730E 01	0.10721E 01	0.20597E 00
0.12444E 01	0.13273E 01	0.10337E 01	0.24351E 00
0.14222E 01	0.12770E 01	0.10595E 01	0.20741E 00
0.15999E 01	0.11700E 01	0.10462E 01	0.12378E 00
0.17777E 01	0.10509E 01	0.10239E 01	0.27038E -01
0.19555E 01	0.95434E 00	0.10074E 01	-0.53109E -01
0.21333E 01	0.89647E 00	0.99771E 00	0.99242E -01
0.23110E 01	0.88561E 00	0.99353E 00	-0.10791E 00
0.24888E 01	0.90623E 00	0.99303E 00	-0.86798E -01
0.26665E 01	0.94477E 00	0.99444E 00	-0.49473E -01
0.28443E 01	0.93636E 00	0.99644E 00	-0.10074E -01
0.30221E 01	0.10139E 01	0.99825E 00	0.20655E -01
0.31998E 01	0.10368E 01	0.99953E 00	0.37267E -01
0.33776E 01	0.10398E 01	0.10003E 01	0.39542E -01
0.35554E 01	0.10316E 01	0.10006E 01	0.31044E -01
0.37331E 01	0.10177E 01	0.10006E 01	0.17095E -01
0.39109E 01	0.10033E 01	0.10004E 01	0.28772E -02
0.40887E 01	0.99239E 00	0.10003E 01	-0.78706E -02
0.42664E 01	0.99671E 00	0.10001E 01	-0.13405E -01
0.44442E 01	0.98616E 00	0.10000E 01	-0.13848E -01
0.46219E 01	0.98938E 00	0.99996E 00	-0.10593E -01
0.47997E 01	0.99436E 00	0.99994E 00	0.55901E -02
0.49775E 01	0.99531E 00	0.99994E 00	-0.62704E -03
0.51552E 01	0.10030E 01	0.99993E 00	0.30069E -02
0.53330E 01	0.10047E 01	0.99997E 00	0.47802E -02
0.55108E 01	0.10048E 01	0.99993E 00	0.47335E -02
0.56885E 01	0.10035E 01	0.99999E 00	0.35464E -02
0.58663E 01	0.10018E 01	0.99999E 00	0.17658E -02
0.60441E 01	0.10000E 01	0.99999E 00	0.58770E -04
0.62218E 01	0.99993E 00	0.99999E 00	-0.11539E -02
0.63996E 01	0.99923E 00	0.99999E 00	-0.17039E -02
0.65774E 01	0.99834E 00	0.99999E 00	-0.16510E -02
0.67551E 01	0.99880E 00	0.99999E 00	-0.11871E -02
0.69329E 01	0.99943E 00	0.99999E 00	-0.55367E -03
0.71107E 01	0.10000E 01	0.99999E 00	0.31114E -04
0.72884E 01	0.10004E 01	0.99999E 00	0.43261E -03
0.74662E 01	0.10006E 01	0.99999E 00	0.59950E -03
0.76440E 01	0.10003E 01	0.99999E 00	0.36040E -03
0.78217E 01	0.10004E 01	0.99999E 00	0.38633E -03
0.79995E 01	0.10002E 01	0.99999E 00	0.16483E -03



S SEVENTH ORDER VS REDUCED ORDER EQUATIONS RUN 1 POSIT VS TIME

THIS IS SEVENTH-ORDER VS REDUCED-ORDER EQUATIONS

3 RUNS ARE CALLED FOR

INPUT DATA RECORD FOR RUN NUMBER 2

ORDER OF EQUATIONS = 19
 INITIAL TIME = 0.0
 FINAL TIME = 0.8000E 01
 STEP SIZE = 0.2885E-02

THE ONLY NON-ZERO CONSTANT IS
 C(1) = 0.1000E 01

ALL THE INITIAL CONDITIONS ARE ZERO

THE COLUMN HEADINGS AND THE CORRESPONDING VARIABLES ARE

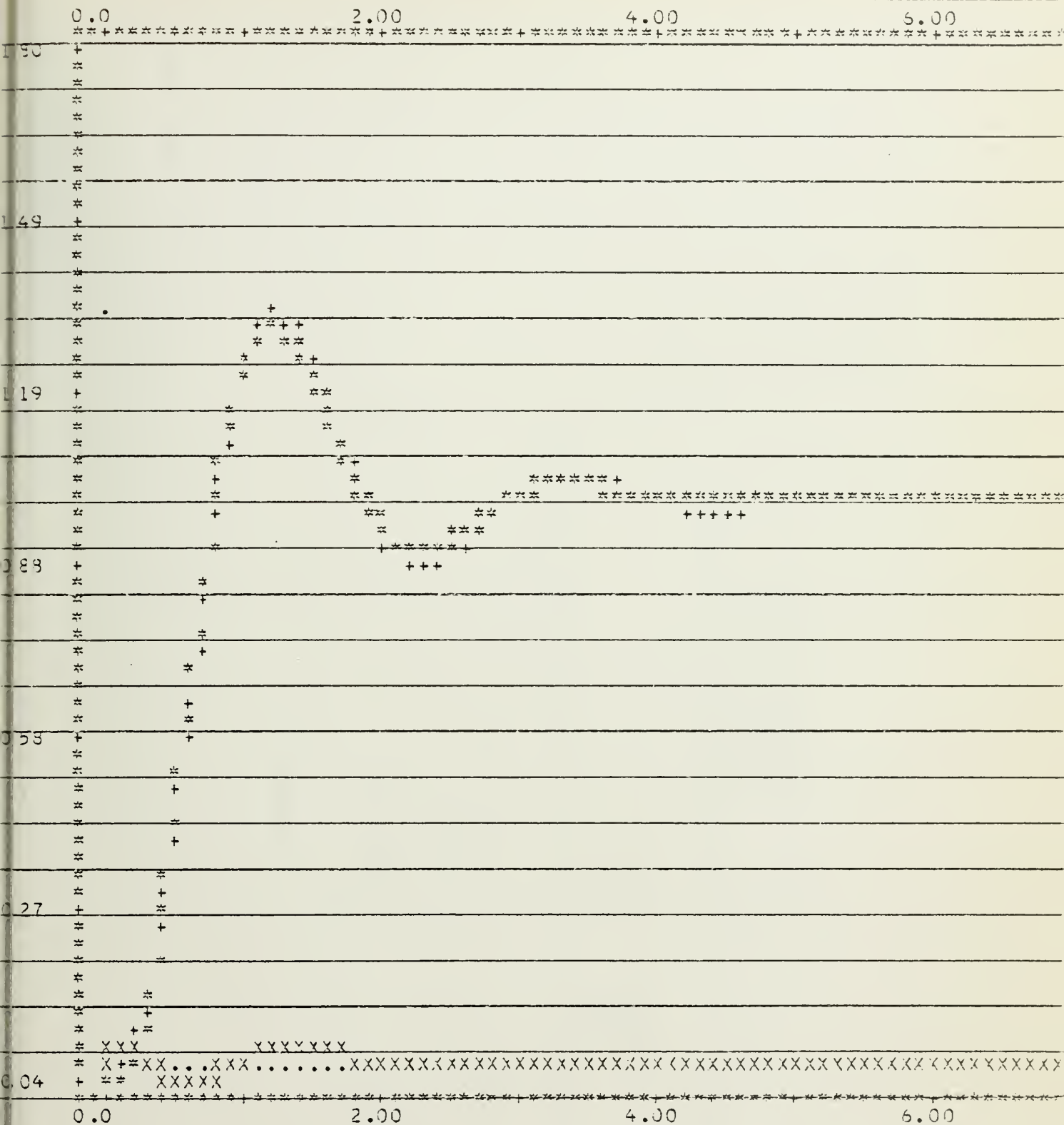
TIME	X(0)
DRIG	X(20)
THIRD	X(22)
ERROR	X(25)

THE GRAPH TITLE AND THE CORRESPONDING VARIABLES ARE

PCSI1 VS TIME	X(20) VS. X(0)
	X(22) VS. X(0)
	X(25) VS. X(0)

THESES SEVENTH ORDER VS REDUCED ORDER EQUATIONS

TIME		CRIS		THIRD		ERROR
0.0		0.0		0.0		0.0
0.177778E	00	0.19093E-01	-01	-0.83620E-02	-02	0.27455E-01
0.355555E	00	0.17544E	00	0.18853E	00	-0.12085E-01
0.533333E	00	0.47702E	00	0.511171E	00	-0.34690E-01
0.711110E	00	0.81911E	00	0.85034E	00	-0.31228E-01
0.888888E	00	0.11630E	01	0.111170E	01	-0.11942E-01
0.106666E	01	0.12730E	01	0.12678E	01	0.10244E-01
0.124444E	01	0.13273E	01	0.13013E	01	0.25443E-01
0.142222E	01	0.12770E	01	0.12473E	01	0.29153E-01
0.159999E	01	0.11700E	01	0.11473E	01	0.22139E-01
0.177777E	01	0.10509E	01	0.10420E	01	0.39540E-02
0.195555E	01	0.93434E	00	0.95932E	00	-0.43812E-02
0.213332E	01	0.86847E	00	0.91348E	00	-0.15013E-01
0.231110E	01	0.83561E	00	0.90439E	00	-0.13778E-01
0.248889E	01	0.90623E	00	0.92262E	00	-0.16339E-01
0.266665E	01	0.94437E	00	0.95473E	00	-0.33155E-02
0.284443E	01	0.93636E	00	0.98820E	00	-0.12374E-02
0.302221E	01	0.10139E	01	0.10139E	01	0.48620E-02
0.319998E	01	0.10368E	01	0.10279E	01	0.39149E-02
0.337776E	01	0.10398E	01	0.10303E	01	0.35568E-02
0.35554E	01	0.10316E	01	0.10241E	01	0.74644E-02
0.373331E	01	0.10177E	01	0.10133E	01	0.38471E-02
0.391109E	01	0.10033E	01	0.10033E	01	0.48730E-04
0.408887E	01	0.99239E	00	0.99525E	00	-0.28617E-02
0.426664E	01	0.98671E	00	0.99102E	00	-0.43077E-02
0.444442E	01	0.98616E	00	0.99043E	00	-0.42637E-02
0.462219E	01	0.98361E	00	0.99247E	00	-0.31076E-02
0.479997E	01	0.99436E	00	0.99573E	00	-0.14255E-02
0.497775E	01	0.99931E	00	0.99911E	00	0.20200E-03
0.515552E	01	0.10030E	01	0.10013E	01	0.13579E-02
0.533330E	01	0.10047E	01	0.10029E	01	0.18625E-02
0.551108E	01	0.10048E	01	0.10030E	01	0.17433E-02
0.568885E	01	0.10035E	01	0.10023E	01	0.11950E-02
0.586663E	01	0.10018E	01	0.10013E	01	0.47493E-03
0.604441E	01	0.10000E	01	0.10002E	01	0.13311E-03
0.622218E	01	0.99883E	00	0.99940E	00	-0.62257E-03
0.639996E	01	0.99828E	00	0.99907E	00	-0.78372E-03
0.657774E	01	0.99834E	00	0.99904E	00	-0.70357E-03
0.675551E	01	0.99880E	00	0.99927E	00	-0.46521E-03
0.693329E	01	0.99843E	00	0.99961E	00	-0.17101E-03
0.711107E	01	0.10000E	01	0.99994E	00	0.93208E-04
0.728884E	01	0.10004E	01	0.10002E	01	0.24605E-03
0.746662E	01	0.10006E	01	0.10003E	01	0.29504E-03
0.764440E	01	0.10005E	01	0.10003E	01	0.25272E-03
0.782217E	01	0.10004E	01	0.10002E	01	0.15354E-03
0.799995E	01	0.10002E	01	0.10001E	01	0.40054E-04



X-SCALE: "*" = 0.100E 00 UNITS

Y-SCALE: "*" = 0.306E-01 UNITS

THESES SEVENTH ORDER VS REDUCED ORDER EQUATIONS RUN 2 POSIT VS TIME

THESIS-SEVENTH ORDER-VS-REDUCED-ORDER-EQUATIONS
3 RUNS ARE CALLED FOR

INPUT DATA RECORD FOR RUN NUMBER 3

ORDER OF EQUATIONS = 19
INITIAL TIME = 0.0
FINAL TIME = 0.8000E-01
STEP SIZE = 0.8889E-02

THE ONLY NON-ZERO CONSTANT IS
C(1) = 0.1000E 01

ALL THE INITIAL CCNDITIONS ARE ZERO

THE COLUMN HEADINGS AND THE CORRESPONDING VARIABLES ARE

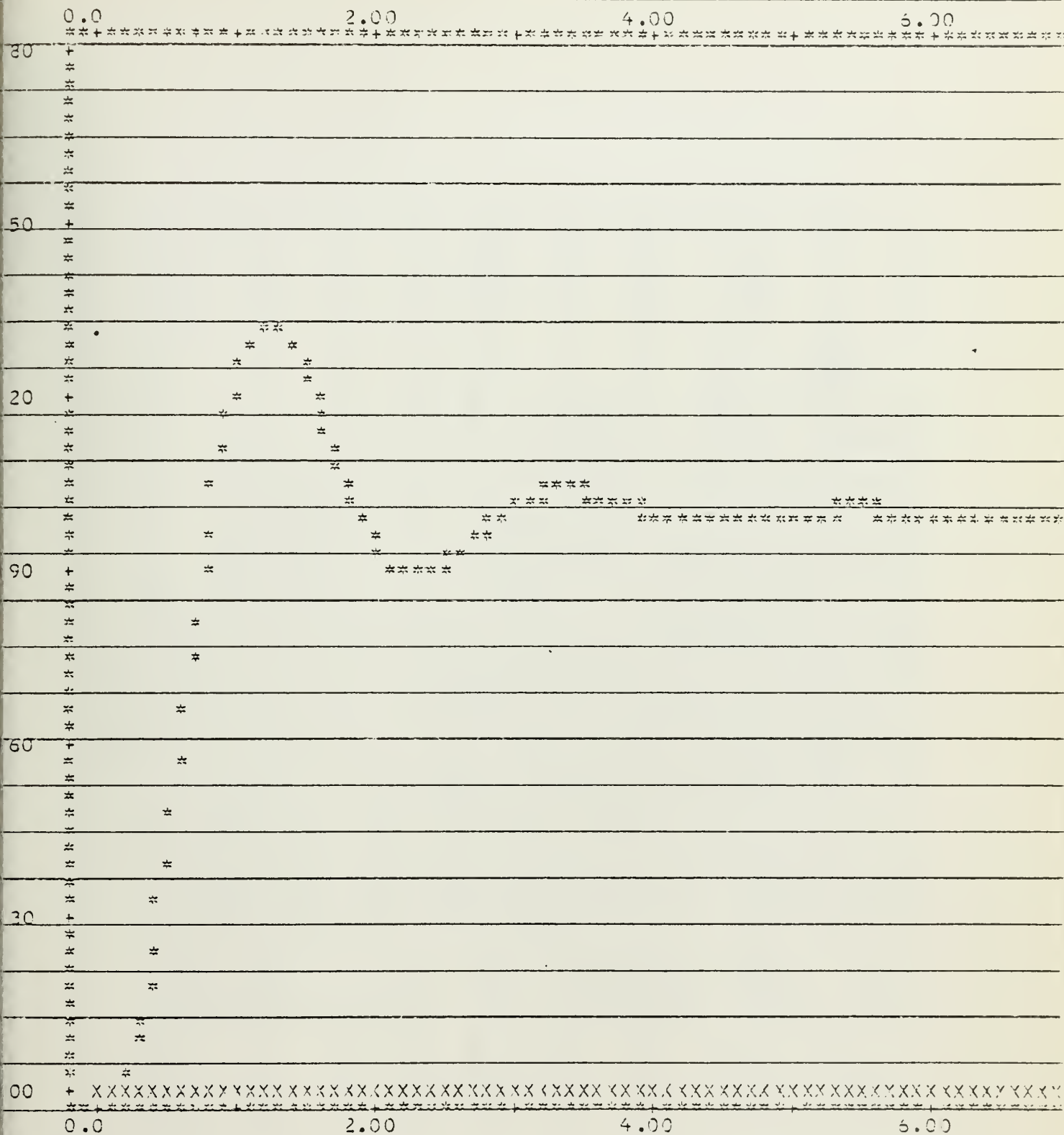
TIME	X(0)
ORIG	X(20)
FOURTH	X(23)
ERROR	X(26)

THE GRAPH TITLE AND THE CORRESPONDING VARIABLES ARE

PCSI1 VS TIME	X(20) VS. X(0)
	X(23) VS. X(0)
	X(26) VS. X(0)

THESIS SEVENTH ORDER VS REDUCED ORDER EQUATIONS

TIME	ERIC	FOURTH	ERROR
0.0	0.0	0.0	0.0
0.177778E C0	0.190933E-01	0.13334E-01	0.70835E-03
0.355555E C0	0.17644E C0	0.17733E C0	-0.10886E-02
0.533333E C0	0.47732E C0	0.47784E C0	-0.32082E-03
0.711110E C0	0.31911E C0	0.31931E C0	-0.20128E-03
0.888888E C0	0.11356E C1	0.11343E C1	0.25272E-03
0.10666E C1	0.12730E C1	0.12770E C1	0.42250E-03
0.12444E C1	0.13273E C1	0.13263E C1	0.49496E-03
0.14222E C1	0.12770E C1	0.12765E C1	0.34714E-03
0.15999E C1	0.11700E C1	0.11693E C1	0.11826E-03
0.17777E C1	0.10509E C1	0.10511E C1	-0.10681E-03
0.19555E C1	0.95434E C0	0.95460E C0	-0.26017E-03
0.21332E C1	0.89647E C0	0.89377E C0	0.30959E-03
0.23110E C1	0.88561E C0	0.88537E C0	-0.25809E-03
0.24888E C1	0.90623E C0	0.90633E C0	-0.14710E-03
0.26665E C1	0.94497E C0	0.94499E C0	-0.28253E-04
0.28443E C1	0.93636E C0	0.93629E C0	0.71704E-04
0.30221E C1	0.10139E C1	0.10182E C1	0.12779E-03
0.31998E C1	0.10368E C1	0.10367E C1	0.13161E-03
0.33776E C1	0.10398E C1	0.10337E C1	0.95367E-04
0.35554E C1	0.10316E C1	0.10316E C1	0.39101E-04
0.37331E C1	0.10177E C1	0.10177E C1	-0.17166E-04
0.39109E C1	0.10033E C1	0.10034E C1	-0.59128E-04
0.40887E C1	0.99239E C0	0.99247E C0	-0.78201E-04
0.42664E C1	0.98671E C0	0.98673E C0	-0.73314E-04
0.44442E C1	0.98616E C0	0.98622E C0	-0.54240E-04
0.46219E C1	0.98936E C0	0.98939E C0	-0.27537E-04
0.47997E C1	0.99436E C0	0.99436E C0	-0.36359E-05
0.49775E C1	0.99921E C0	0.99930E C0	0.11931E-04
0.51552E C1	0.10030E C1	0.10029E C1	0.19073E-04
0.53330E C1	0.10047E C1	0.10047E C1	0.17166E-04
0.55108E C1	0.10048E C1	0.10048E C1	0.26294E-05
0.56885E C1	0.10035E C1	0.10035E C1	-0.57220E-05
0.58663E C1	0.10018E C1	0.10018E C1	-0.13120E-04
0.60441E C1	0.10009E C1	0.10009E C1	-0.24730E-04
0.62218E C1	0.99833E C0	0.99836E C0	-0.27359E-04
0.63996E C1	0.99828E C0	0.99831E C0	-0.22292E-04
0.65774E C1	0.99834E C0	0.99835E C0	-0.17235E-04
0.67551E C1	0.99830E C0	0.99831E C0	-0.11683E-04
0.69329E C1	0.99643E C0	0.99944E C0	-0.59009E-05
0.71107E C1	0.10000E C1	0.10000E C1	-0.38147E-05
0.72884E C1	0.10004E C1	0.10004E C1	-0.33147E-05
0.74662E C1	0.10006E C1	0.10006E C1	-0.47634E-05
0.76440E C1	0.10005E C1	0.10006E C1	-0.76294E-05
0.78217E C1	0.10004E C1	0.10004E C1	-0.95357E-05
0.79995E C1	0.10002E C1	0.10002E C1	-0.10490E-04



X-SCALE: "*" = 0.100E 00 UNITS

Y-SCALE: "*" = 0.300E-01 UNITS

SIS SEVENTH ORDER VS REDUCED ORDER EQUATIONS RUN 3 POSIT VS TIME


```

CARD COLUMN NR      DESCRIPTION
1 1-20      PROBLEM IDENTIFICATION
2 21-22      IREQ=1 REDUCED EQUATIONS ONLY
                IREQ=2 REDUCED EQN'S + TIME RESPONSE
                L=ORDER OF ORIGINAL EQUATION
23-24
1-12      NUMERATOR COEFFICIENTS IN ASCENDING
13-24      POWERS OF S. MUST HAVE NUMBER OF
                CARDS TO COVER ORDER OF EQUATION
                ETC
3 1-12      DENOMINATOR COEFFICIENTS IN ASCENDING
13-24      POWERS OF S.
                ETC
4 1-12      FREQ=FREQUENCY DESIGNATED FOR SINUSOIDAL
                INPUT IN TIME RESPONSE.
                E12.5
5 1-48      GRAPH TITLE. APPEARS ON GRAPHS.
                10A8
6 1      NUMBER OF RUNS TO BE PROCESSED (0 LE. 5)
                11
7 1-10      INITIAL VALUE OF TIME
11-20      INTEGRATION STEP SIZE
21-30      FINAL VALUE OF TIME
                (ALL PUNCHED
                WITH DECIMAL
                POINTS)
                8F10.4
8 1-10      VALUES OF CONSTANTS
                STEP INPUT OF 1.0
                C(1)=1.0 WILL GIVE RAMP INPUT OF 1
                C(2)=1.0 WILL GIVE SINUSOIDAL INPUT CF FREQ
                C(3)=1.0 WILL GIVE SIN(OMEGA * T) WHERE OMEGA=2*PI*FREQ
                SIN(OMEGA * T)
                8F10.4
9 1-10      INITIAL CONDITION VALUES OF VARIABLES
11-20      PUNCHED WITH DECIMAL POINTS
                ORDER
                X(20)=GRIG EQN, X(21)=SECOND
                X(22)=THIRD ORDER, AND X(23)=FOURTH ORDER
                USE THREE BLANK CARDS IF ALL IC'S ARE ZERO.
10 1-10      CHOICE OF VARIABLES FOR PRINT OUT IN
                TABLE FORM. EACH GROUP OF 10 COLUMNS MAY
                SPECIFY A TABLE HEADING (8 CHAR) AND A
                2 DIGIT, RIGHT JUSTIFIED SUBSCRIPT TO
                8(A8,I2)

```



```

*****
BE PRINTED UNDER 'ORIG' VS 'TIME'.
8 VARIABLES MAXIMUM FOR ANY ONE RUN.
FOR THIS PROGRAM, THE FOLLOWING WILL
GIVE THE ORIGINAL AND REDUCED ORDER EQN'S
VS TIME FOR RUNS 1, 2, AND 3:
1 TIME 00 ORIG 20 SECOND 21 ERROR 24
2 TIME 00 ORIG 20 THIRD 22 ERROR 25
3 TIME 00 ORIG 20 FOURTH 23 ERROR 26
*****
11 CHOICE OF VARIABLES FOR GRAPHICAL OUTPUT 4(2A8,2I2)
UP TO 4 CURVES MAY BE PLOTTED, SEPARATELY
OR ALL ON ONE GRAPH.
EX: POSIT VS TIME 2000 IN CCL(1-20) WILL
CAUSE X(20) TO BE PLOTTED AGAINST TIME
AND LABELED ORIG VS TIME. THE FIRST 16
COLUMNS FOR LABEL AND COLUMNS(17-18) FOR
Y COORDINATE AND 19-20 FOR X COORDINATE.
FOR MULTIPLE PLOTS ON SINGLE GRAPH,
ONLY THE FIRST TITLE REQUIRED. THE DATA CARDS
NUMBERS 7 THROUGH 11 NEED TO BE SUPPLIED
FOR EACH ADDITIONAL RUN.
*****

```

```

*****
NOTE: INTEGRATION STEP SIZE IS A GUESSTIMATE
AND BEST FOUND BY TRIAL AND ERROR.
DECK BLANK CARDS MUST BE INCLUDED IN DATA
INITIAL CONDITION STATE VALUES ARE:
ORIG EQN: X(1)--X(10) CORRESPOND TO X(1)-X(10)
SECOND ORDER: X(11)-X(12) CORRESPOND TO X(1)-X(2)
THIRD ORDER: X(13)-X(15) CORRESPOND TO X(1)-X(3)
FOURTH ORDER: X(16)-X(19) CORRESPOND TO X(1)-X(4)
*****

```

```

*****
PROGRAMMER: J. D. THOMPSON
LT, USN JUNE 76
*****
DIMENSION A(4),B(4),W(12),Y(12),U(8),Z(9),X(26),D(15),CP(12),
1XCD(19),C(15),CN(12),CD(12),RTR(20),RTI(20),NAME(5),POL(12)
C(10)=1.
777 FCRMAT(1.1)
901 FCRMAT(/6X,30HROOTS OF DENOMINATOR OF ORDER ,I1,2X,7HERROR, I2,/8X
901 *,SHREAL PART,8X,10HIMAG. PART)
907 FCRMAT(5X,E13.6,7X,E13.6)
2001 FCRMAT(5A4,2I2)
2002 FCRMAT(4E12.5)
2003 FCRMAT (1P6E20.7)
*****

```



```

K=I-4
J=K+K
W(I)=CC(J)-A(2)*W(K)
2  Y(I)=CN(J)-B(2)*W(K)
  A(3)=W(1)/W(6)
  B(3)=Y(1)/W(6)
3  IF(L-4)140,3,3
  CC 4 I=10,12
K=I-3
J=K-5
W(I)=W(J)-A(3)*W(K)
4  Y(I)=W(J)-B(3)*W(K)
  A(4)=W(6)/W(10)
  B(4)=Y(6)/W(10)
GO TO 140
140 Z(1)=1.
  Z(2)=A(1)*A(2)
  Z(3)=A(1)+A(3)
  Z(4)=A(2)*A(3)
  Z(5)=Z(2)*A(3)
  Z(6)=A(2)+A(4)
  Z(7)=Z(2)+A(4)*Z(3)
  Z(8)=A(4)*Z(4)
  Z(9)=A(1)*Z(8)
  U(1)=A(2)*B(1)
  U(2)=B(1)+B(3)
  U(3)=A(3)*B(2)
  U(4)=Z(4)*B(1)
  U(5)=B(2)+B(4)
  U(6)=B(1)*A(2)+A(4)*B(2)
  U(7)=A(3)*A(4)*B(2)
  U(8)=Z(8)*B(1)
PRINT 5007
ICRD=1
PRINT 5010,IORD
WRITE(6,5011)B(1)
WRITE(6,5013)A(1),Z(1)
ICRD=2
PRINT 5010,IORD
WRITE(6,5011)U(1),B(2)
WRITE(6,5013)Z(2),A(2),Z(1)
ICRD=3
PRINT 5010,IORD
WRITE(6,5011)U(4),U(3),U(2)
WRITE(6,5013)Z(5),Z(4),Z(3),Z(1)
ICRD=4
PRINT 5010,IORD
WRITE(6,5011)U(8),U(7),U(6),U(5)

```



```

WRITE(6,5013)Z(9),Z(8),Z(7),Z(6),Z(1)
KCLNT=0
LC=L+1
DC 899 I=1,12
CP(I)=CC(I)
GC TO 900
900 KCLNT=KCLNT + 1
CALL PRGD(CP,LD,RTR,RTI,POL,IR,IER)
IF(KCLNT - 4)299,301,302
299 IF(KCLNT - 2)300,302,303
300 KCRD=L
WRITE(6,901)KORD,IER
DO 77 I=1,L
77 PRINT 907,RTR(I),RTI(I)
LC=3
CP(1)=Z(2)
CP(2)=A(2)
CP(3)=Z(1)
GC TO 900
302 KCRD=2
WRITE(6,901)KORD,IER
DC 78 I=1,2
78 PRINT 907,RTR(I),RTI(I)
LC=4
CP(1)=Z(5)
CP(2)=Z(4)
CP(3)=Z(3)
CP(4)=Z(1)
GC TO 900
303 KCRD=3
WRITE(6,901)KORD,IER
DC 79 I=1,3
79 PRINT 907,RTR(I),RTI(I)
LC=5
CP(1)=Z(9)
CP(2)=Z(8)
CP(3)=Z(7)
CP(4)=Z(6)
CP(5)=Z(1)
GC TO 900
301 KCRD=4
WRITE(6,901)KORD,IER
DC 81 I=1,4
81 PRINT 907,RTR(I),RTI(I)
IF(IREQ - 2)800,801,801

```

C
C THIS CONCLUDES CALCULATIONS OF NUMERATORS AND DENOMINATORS
C FOR THE APPROXIMANTS, INCLUDING PRINT OUT CF VALUES


```

C      IN ACCENDING POWERS OF S AND IN INCREASING CRDER
C      CF APPROXIMANT.
      STOP
      GC TO 801
      CCNTINUE
      PRINT 5015
5015  FCRMAT(/6X,11HERROR CODES,/6X,16HIER=0, NO ERRORS,/6X,45HIER=1, NO
      1 CCNVERGENCE WITH FEASIBLE TOLERANCE,/6X,44HIER=2, POLY IS DEGENER
      2ATE (CONSTANT OR ZERO),/6X,42HIER=3, SUBROUTINE ABANCONEC (ZERO DI
      3VISOR),/6X,27HIER=4, NO S-FRACTION EXISTS,/6X,37HIER=-1, POOR ACCU
      4RACY IN CALCULATIONS)
      DC 499 I=1,15
      C(I)=0
      IF(L-4) 500,501,502
502  IF(L-6) 503,504,505
503  IF(L-8) 506,507,508
508  IF(L-10) 509,510,511
500  D(9)=1
      GC TO 444
501  C(10)=1
      C(I)=1
      GC TO 444
503  D(11)=1
      C(4)=1
      C(5)=1
      GC TO 444
504  D(12)=1
      DC 512 I=1,3
512  C(I)=1
      GC TO 444
506  D(13)=1
      DC 513 I=1,4
513  C(I)=1
      GC TO 444
507  C(14)=1
      CC 514 I=1,5
514  C(I)=1
      GC TO 444
509  D(15)=1
      DC 515 I=1,6
515  C(I)=1
      GC TO 444
510  C(8)=1
      CC 516 I=1,7
516  C(I)=1
      GC TO 444
511  PRINT 601
601  FCRMAT(5X,'ORDER EXCEEDS 10, VERIFY L VALUE',/)

```



```

100 FCRMAT (10A8)
101 READ (5,101)NR
102 FCRMAT (11)
103 NN=19
104 NRC = 0
105 GC TO 1000
106 NRC = NRC + 1
107 WRITE (6,201) (ITITLE(I),I=1,6)
201 FCRMAT (1H1,///,36X,6A8)
108 IF(NRC.EQ.1.AND.NR.EQ.1) GO TO 5
202 WRITE(6,202)NR
203 FCRMAT (/,37X,11,20H RUNS ARE CALLED FOR )
204 GC TO 6
205 WRITE(6,203)
206 FCRMAT (/,37X,21H ONE RUN IS CALLED FOR ,///,18H INPUT DATA RECORD)
207 GC TO 7
208 WRITE(6,204)NRC
209 FCRMAT (///,34H INPUT DATA RECORD FOR RUN NUMBER ,11)
210 WRITE(6,205)NN
211 FCRMAT (///,22H ORDER OF EQUATIONS = ,I2)
212 READ (5,103)TI,DT,TF1,DT2,TF2,DT3,TF3
103 FCRMAT (8F10.4)
104 TF = TF1
105 IF(DT2.NE.0.) GO TO 9
106 WRITE(6,206)TI,TF
107 FCRMAT (22H INITIAL TIME = ,E10.4, /
108 WRITE(6,207)DT
109 FCRMAT (22H FINAL TIME = ,E10.4)
110 WRITE(6,208)DT
111 FCRMAT (22H STEP SIZE = ,E10.4)
112 GC TO 12
113 IF(DT3.NE.0.) GO TO 11
114 TF = TF2
115 WRITE(6,209)TI,TF
116 WRITE(6,208)DT,TF1,DT2,TF1,TF
117 FCRMAT (22H STEP SIZE = ,E10.4,13H BETWEEN / = ,E10.4,
118 GO TO 12
119 TF=TF3
120 WRITE(6,208) DT,TF1,TF1,DT2,TF1,TF2,TF2,TF
121 READ(5,103) (C(I),I=1,8)
122 READ (5,103)(X(I),I=1,NN)
123 J = 0
124 DC 14 I=1,8
125 IF(C(I).NE.0.) J=J+1
126 CCATINUE
127 K = 0
128 DC 16 I=1,NN
129 IF(X(I).NE.0.) K=K+1

```

```

INT12700
INT12710
INT12720
INT12750
INT12800
INT12810
INT12820
INT12830
INT12840
INT12850
INT12860
INT12870
INT12880
INT12890
INT12900
INT12910
INT12920
INT12930
INT12940
INT12950
INT12960
INT12970
INT12980
INT12990
INT13000
INT13010
INT13020
INT13030
INT13040
INT13050
INT13060
INT13070
INT13080
INT13090
INT13100
INT13110
INT13120
INT13130
INT13140
INT13150
INT13160
INT13170
INT13180
INT13190
INT13200
INT13210

```



```

16 CCNTINUE
17 IF(J-1)17,18,19
205 WRITE(6,209)
    FORMAT(/,34H ALL THE CONSTANTS, C(I), ARE ZERO )
18 GC TO 423
210 WRITE(6,210)
    FORMAT(/,30H THE ONLY NON-ZERO CONSTANT IS )
19 GC TO 420
211 WRITE(6,211)
    FORMAT(/,35H THE NON-ZERO CONSTANTS, C(I), ARE )
420 CC 422 I=1,8
212 IF(C(I).NE.0.) WRITE(6,212) I,C(I)
422 CCNTINUE
423 IF(K-1)424,425,426
424 WRITE(6,1209)
1209 FCRMAT(/,36H ALL THE INITIAL CONDITIONS ARE ZERO )
    GC TO 20
425 WRITE(6,1210)
1210 FCRMAT(/,39H THE ONLY NON-ZERO INITIAL CONCITION IS )
    GC TO 427
426 WRITE(6,1211)
1211 FCRMAT(/,36H THE NON-ZERO INITIAL CONDITIONS ARE )
427 DC 429 I=1,NN
1212 IF(X(I).NE.0.) WRITE(6,1212) I,X(I)
429 FCRMAT(14X,2HX(,12,4H) = ,E10.4)
20 READ(5,104) (JTITLE(I),IP(I),I=1,8)
104 FORMAT(8(A8,12))

C CHECK FOR THE NUMBER OF COLUMNS CALLED FOR BY LOCATING FIRST
C BLANK COLUMN HEADING
C
    DO 21 J=1,8
    IF(JTITLE(J).EQ.IBLANK) GO TO 22
21 CCNTINUE
    J=9
22 JJ = J - 1

C JJ IS NOW THE NUMBER OF COLUMNS. REPEAT WITH THE GRAPHS.
C
    READ(5,105)(KTITLE(I),KTITLE(I+1),IG(I),IG(I+1),I=1,7,2)
105 FCRMAT(4(2A8,2I2))
    CC 24 K=1,7,2
24 IF(KTITLE(K).EQ.IBLANK.AND.KTITLE(K+1).EQ.IBLANK) GO TO 25
    CCNTINUE
    K=8
25 KK = K/2

```

```

INT113220
INT113230
INT113240
INT113250
INT113260
INT113270
INT113280
INT113290
INT113300
INT113310
INT113320
INT113330
INT113340
INT113350
INT113360
INT113370
INT113380
INT113390
INT113400
INT113410
INT113420
INT113430
INT113440
INT113450
INT113460
INT113470
INT113480
INT113490
INT113500
INT113510
INT113520
INT113530
INT113540
INT113550
INT113560
INT113570
INT113580
INT113590
INT113600
INT113610
INT113620
INT113630
INT113640
INT113650
INT113660
INT113670
INT113680
INT113690

```



```

KKK = KK*2
MULTIP = 0
IF (KK.NE.1) GO TO 306
IF (IG(3) + IG(4).EQ.0) GO TO 306
IF (IG(5) + IG(6).NE.0) GO TO 303
MULTIP = 2
KKK = 4
GC TO 306
303 IF (IG(7) + IG(8).NE.0) GO TO 305
MULTIP = 3
KKK = 6
GC TO 306
305 MULTIP = 4
KKK = 8

IF MULTIP = 0, KK IS THE NUMBER OF SINGLE CURVE GRAPHS. OTHERWISE
MULTIP IS THE NUMBER OF CURVES ON A SINGLE GRAPH.

306 IF (JJ.EQ.0) GO TO 27
WRITE(6,214) (JTITLE(I),IP(I),I=1,JJ)
214 FCFORMAT (///,56H THE COLUMN HEADINGS AND THE CORRESPONDING VARIABLE
1S ARE ,/(10X,A8,4X,2HX(,I2,1H))),
GC TO 28
27 WRITE(6,215)
215 FCFORMAT (///,25H NO PRINTOUT IS REQUIRED )
28 IF (KK.EQ.0) GO TO 308
IF (MULTIP.NE.0) GO TO 309
IF (KK.NE.1) GO TO 307
WRITE(6,216) KTITLE(1),KTITLE(2),IG(1),IG(2)
216 FCFORMAT (///,52H THE GRAPH TITLE AND THE CORRESPONDING VARIABLES AR
1E ,/(10X,2A8,4X,2HX(,I2,8H) VS. X(I2,1H))
GC TO 31
307 WRITE(6,217) (KTITLE(1),KTITLE(I+1),IG(I),IG(I+1),I=1,KKK,2)
217 FCFORMAT (///,64H THE INDIVIDUAL GRAPH TITLES AND THE CORRESPONDING
1VARIABLES ARE ,/(10X,2A8,4X,2HX(,I2,8H) VS. X(I2,1H)))
GC TO 31
308 WRITE(6,1217)
1217 FCFORMAT (///,24H NO GRAPHS ARE REQUIRED )
GC TO 31
309 WRITE(6,1220)
1220 FCFORMAT (///,52H THE GRAPH TITLE AND THE CORRESPONDING VARIABLES AR
1E ,/
WRITE(6,1221) KTITLE(1),KTITLE(2), (IG(1),IG(I+1),I=1,KKK,2)
1221 FCFORMAT (10X,2A8,4X,2HX(,I2,8H) VS. X(I2,1H)), (30X,2HX(,I2,
8H) VS. X(I2,1H)))

THIS ENDS THE BOOK-KEEPING. INITIALIZE BEFORE ENTERING MAIN LOOP.

```



```

31 IFAGE = 0
   T = T + 1
   NCPTS = 0
   NUMPTS = 0
   ITITLE(8) = IBLANK
   ITITLE(11) = IBLANK
   ITITLE(12) = IBLANK
   RUN(2) = BT(NRC)
   C(11) = 20.
   C(12) = 5.
   C(13) = DT
   DC 42 I = 1, NN
   XC(I) = X(I)
42 TC = T
   C(10) = 2.
   RETURN
C
C 2000 IF(JJ.EC.0) GO TO 54
      INCPR = C(11)+0.0000001
      C(11) = 20.
      IF( MOD (NOPTS, 50*INCPR).EQ.0) GO TO 46
      IF( MOD (NOPTS, 10*INCPR).EQ.0) GC TO 47
      IF( MOD (NOPTS, INCPR)) 54, 48, 54
46 IFAGE = IPAGE + 1
      IF(NR.EQ.1) GO TO 1047
      WRITE(6, 218) (ITITLE(I), I=1, 6), IPAGE, ITITLE(7), (JTITLE(I), I=1, 8)
      WRITE(6, 219)
      GC TO 47
1047 WRITE(6, 1218) (ITITLE(I), I=1, 6), IPAGE, (JTITLE(I), I=1, 8)
      WRITE(6, 219)
47 WRITE(6, 219)
218 FCORMAT (1H1, ///, 20X, 6A8, 10X, 5HPAGE , 11, 14H CF OUTPUT FOR, A8, ///, //,
1218 1 FFORMAT (1H1, ///, 20X, 6A8, 30X, 5HPAGE , 11, ///, //, 11X, 8(A8, 5X))
219 FFORMAT (1H1, //, 20X, 6A8, 5X)
48 DC 49 I = 1, NN
49 XC(I) = X(I)
   TC = T
   C(10) = 3.
   RETURN
C
C 50 DC 53 I = 1, JJ
C
      PR(I) = T
      IF(IP(I).NE.0) PR(I)=XC(IP(I))
53 CCNTINUE
      WRITE(6, 220) (PR(I), I=1, JJ)

```

```

INT 14180
INT 14190
INT 14200
INT 14210
INT 14220
INT 14230
INT 14240
INT 14250
INT 14260
INT 14270
INT 14280
INT 14290
INT 14300
INT 14310
INT 14320
INT 14330
INT 14340
INT 14350
INT 14360
INT 14370
INT 14380
INT 14390
INT 14400
INT 14410
INT 14420
INT 14430
INT 14440
INT 14450
INT 14460
INT 14470
INT 14480
INT 14490
INT 14500
INT 14510
INT 14520
INT 14530
INT 14540
INT 14550
INT 14560
INT 14570
INT 14580
INT 14590
INT 14600
INT 14610
INT 14620
INT 14630
INT 14640
INT 14650

```



```

220 FORMAT (7X, 8E13.5)
54 IF(KK.EQ.0) GO TO 62
   INCGR = C(12)+0.0000001
   C(12) = 5
   IF( MOD (NOPTS, INCGR).NE.0) GO TO 62
57 DC 57 I=1,NN
   XC(I) = X(I)
   TC = T
   C(10) = 4.
   RETURN
C
58 DC 61 I=1,KKK
C
   GR(I) = T
   IF(IG(I).NE.0) GR(I)=XC(IG(I))
61 CCNTINUE
   IF(KKK.GE.8) GO TO 1610
   KP1 = KKK + 1
   DC 1612 I=KP1,8
1612 GR(I) = 0.
1610 NUMPTS = NUMPTS + 1
   Y1(NUMPTS) = GR(1)
   X1(NUMPTS) = GR(2)
   Y2(NUMPTS) = GR(3)
   Y3(NUMPTS) = GR(4)
   Y3(NUMPTS) = GR(5)
   X3(NUMPTS) = GR(6)
   Y4(NUMPTS) = GR(7)
   X4(NUMPTS) = GR(8)
62 NCPTS = NOPTS + 1
   IF(NUMPTS.LT.900) GO TO 64
   WRITE (6,221)
221 FCORMAT (//////,25H STOP AT 900 GRAPH POINTS )
64 IF(NOPTS.LT.4500) GO TO 66
   WRITE (6,222)
222 FCORMAT (//////,31H STOP AT 4500 INTEGRATION STEPS )
66 GO TO 91
66 IF(IPAGE - 9)69,67,68
67 IF( MOD (NOPTS , 50*INCPR).NE.0) GO TO 69
68 WRITE (6,223)
223 FCORMAT (//////,27H STOP AT 9 PAGES OF OUTPUT )
69 GO TO 91
69 DC 70 I=1,NN
70 IF(ABS(X(I)).GT.1.E+12) GO TO 71
   CCNTINUE
71 WRITE (6,224)

```

```

INT14660
INT14670
INT14680
INT14690
INT14700
INT14710
INT14720
INT14730
INT14740
INT14750
INT14760
INT14770
INT14780
INT14790
INT14800
INT14810
INT14820
INT14830
INT14840
INT14850
INT14860
INT14870
INT14880
INT14890
INT14900
INT14920
INT14930
INT14940
INT14950
INT14960
INT14970
INT14980
INT14990
INT15000
INT15010
INT15020
INT15030
INT15040
INT15050
INT15060
INT15070
INT15080
INT15090
INT15100
INT15110
INT15120
INT15130

```



```

224 FORMAT (//////,76H STOP WITH THE ABSOLUTE VALUE OF A DEPENDENT VAR INT15140
1TABLE GREATER THAN 1.0E+12.,/,57H INTEGRATION PROBABLY UNSTABLE.,
2 TRY A SMALLER STEP SIZE.,26HNO GRAPHS WILL BE PLOTTED.
GC TO 330 INT15150
CT = C(13) INT15160
IF(TT.GT.TF) GO TO 80 INT15170
IF(T.LT.TF) GO TO 75 INT15180
74 WRITE(6,225) INT15190
225 FCN MAT (//////,26H NORMAL STOP AT FINAL TIME ) INT15200
GC TO 91 INT15210
75 IF(T.GE.TF1) GO TO 77 INT15220
76 C(13) = DT INT15230
GC TO 87 INT15240
77 IF(T.GE.TF2) GO TO 79 INT15250
78 C(13) = DT2 INT15260
GC TO 87 INT15270
79 C(13) = DT3 INT15280
GC TO 87 INT15290
80 IF(TF.GE.T) GO TO 74 INT15300
IF(TF1.LT.T) GO TO 76 INT15310
IF(TF2 - T)78,79,75 INT15320
87 C(10) = 5. INT15330
INT15340
C INT15350
C CALL RKUTTA (NN,T,X,DT,C,TC,XC,DX) INT15360
INT15370
90 IF(C(10).EQ.6.) RETURN INT15380
T = T + DT INT15390
GC TO 2000 INT15400
91 IF(KK.EQ.0) GO TO 330 INT15410
IF(MULTIP.NE.0) GO TO 97 INT15420
C INT15430
C PRINT PLOT UP TO 4 INDIVIDUAL CURVES INT15440
C NUMPTS=-NUMPTS INT15450
C DO 310 II=1,KK INT15460
WRITE(6,9998) INT15470
9998 FORMAT(1H1) INT15480
ITITLE(9)=KTITLE(2*II-1) INT15490
ITITLE(10)=KTITLE(2*II) INT15500
GC TO (311,312,313,314),II INT15510
311 CALL PLOTP(X1,Y1,NUMPTS,0) INT15520
CALL GC TO 310 INT15530
312 CALL PLOTP(X2,Y2,NUMPTS,0) INT15540
GC TO 310 INT15550
313 CALL PLOTP(X3,Y3,NUMPTS,0) INT15560
GC TO 310 INT15570
314 CALL PLOTP(X4,Y4,NUMPTS,0) INT15580
310 WRITE(6,9999) ITITLE INT15590
INT15600
INT15610

```



```

9959 FOFMAT(1H0,8X,12A8)
C
C
C
97
57
PLCT DUMMY CURVE ALONG AXES TO SET SCALES FOR MULTIPLE PLOT
BIGX = 0.
BIGY = 0.
SMLX = 0.
SMLY = 0.
DC 1970 I=1, NUMPTS
XMAX = AMAX1 ( X1(I), X2(I), X3(I), X4(I) )
YMAX = AMAX1 ( Y1(I), Y2(I), Y3(I), Y4(I) )
XMIN = AMIN1 ( X1(I), X2(I), X3(I), X4(I) )
YMIN = AMIN1 ( Y1(I), Y2(I), Y3(I), Y4(I) )
IF (BIGX.LT.XMAX) BIGX=XMAX
IF (BIGY.LT.YMAX) BIGY=YMAX
IF (SMLX.GT.XMIN) SMLX=XMIN
IF (SMLY.GT.YMIN) SMLY=YMIN
CCNTINUE
1970
TX(1) = 0.
TX(2) = 0.
TX(3) = 0.
TX(4) = 0.
TX(5) = SMLX
TY(1) = BIGX
TY(2) = BIGY
TY(3) = SMLY
TY(4) = 0.
TY(5) = 0.
WRITE(6,9998)
ITITLE(5) = KTITLE(1)
ITITLE(10) = KTITLE(2)
NT=-5
CALL PLCTP(TX,TY,NT,1)
MCCUR = 2
DC 410 II=1,MULTIP
IF (II.EQ.MULTIP) MODCUR=3
GO TO (411,412,413,414),II
411 CALL PLCTP(X1,Y1,NUMPTS,MODCUR)
GC TO 410
412 CALL PLCTP(X2,Y2,NUMPTS,MODCUR)
GC TO 410
413 CALL PLCTP(X3,Y3,NUMPTS,MODCUR)
GC TO 410
414 CALL PLCTP(X4,Y4,NUMPTS,MODCUR)
410 CCNTINUE
WRITE(6,9999) ITITLE
C

```



```

330 IF(NRC.NE.NR) GO TO 1000
    IF(NR.GT.1) GO TO 333
    WRITE(6,226)
    FORMAT(/,43H THE ONE RUN CALLED FOR HAS BEEN COMPLETED. ,/)
226 STOP
333 WRITE(6,227)NR
227 FORMAT(/,5H THE ,11,37H RUNS CALLED FOR HAVE BEEN COMPLETED.,/)
    STOP
    ENCL
    SUBROUTINE RKUTTA(/NN/,/T/,/X/,/CT/,/C/,/TC/,/XC/,/DX/)
    DIMENSION X(30), C(15), XC(30), DX(30), CT(4), AK(4,30)
    REAL*8 AK,CT
    INCLC = C(10) - 4.0+0.00000001
    IF(INCLC.GT.1) GO TO 3
    CT(1) = 0.000
    CT(2) = 0.500
    CT(3) = 0.500
    CT(4) = 1.000
    II=0
    7 II=II+1 + CT(II)*DT
    TC = 1,NN
    2 XC(J) = X(J) + CT(II)*AK(II-1, J)
    3 C(10) = 6.0
    RETURN
    4 DC 4 J=1,NN
    AK(II, J) = DT*DX(J)
    IF(II.LT.4) GO TO 7
    5 DO 5 J=1,NN
    X(J)=X(J)+(AK(1,J)+2.0*(AK(2,J)+AK(3,J))+AK(4,J))/6.0
    RETURN
    END
    SUBROUTINE PRQD
    PURPCE
    CALCULATE ALL REAL AND COMPLEX ROOTS CF A GIVEN POLYNCMIAL
    WITH REAL COEFFICIENTS.
    SUBROUTINE PRQD(C,IC,Q,E,POL,IR,IER)
    DIMENSIONED DUMMY VARIABLES
    DIMENSION E(1),Q(1),C(1),POL(1)
    NORMALIZATION OF GIVEN POLYNOMIAL
    TEST OF DIMENSION
    IR CONTAINS INDEX OF HIGHEST COEFFICIENT
    IER=0

```

```

INT16100
INT16110
INT16120
INT16130
INT16140
INT16150
INT16160
INT16170
INT16180
INT16190
INT16200
INT16210
INT16220
INT16230
INT16240
INT16250
INT16260
INT16270
INT16280
INT16290
INT16300
INT16310
INT16320
INT16330
INT16340
INT16350
INT16360
INT16370
INT16380
INT16390
INT16400
INT16410
INT16420
PRQD 40
PRQD 50
PRQD 60
PRQD 70
PRQD 80
PRQD 90
PRQD 620
PRQD 630
PRQD 640
PRQD 650
PRQD 660
PRQD 670
PRQD 680
PRQD 690
PRQD 700

```

CCCCC C C C C C


```

11 PCL(I+1)=EPS*ABS(Q(I+1))+EPS
12 E(I+1)=Q(I+1)*EXPT
C
C      TEST OF REMAINING DIMENSION
13 IF(ISTA-IEND)12,20,60
14 JEND=IEND-1
C
C      COMPUTATION OF S-FRACTION
15 DC 19 I=ISTA,JEND
16 IF(I-ISTA)13,16,13
17 IF(ABS(E(I))-POL(I+1))14,14,16
C
C      THE GIVEN POLYNOMIAL HAS MULTIPLE ROOTS, THE COEFFICIENTS CF
C      THE COMMON FACTOR ARE STORED FROM Q(NSAV) UP TO Q(IR)
18 NSAV=I
19 DC 15 K=I,JEND
20 IF(ABS(E(K))-POL(K+1))15,15,80
21 CCNTINUE
22 GOTO 21
C
C      EUCLIDEAN ALGORITHM
23 DC 19 K=I,IEND
24 E(K+1)=E(K+1)/E(I)
25 Q(K+1)=E(K+1)-Q(K+1)
26 IF(K-I)18,17,18
C
C      TEST FOR SMALL DIVISOR
27 IF(ABS(Q(I+1))-POL(I+1))80,80,19
28 Q(K+1)=Q(K+1)/Q(I+1)
29 PCL(K+1)=POL(K+1)/ABS(Q(I+1))
30 E(K)=Q(K+1)-E(K)
31 CCNTINUE
32 Q(IR)=-Q(IR)
C
C      THE DISPLACEMENT EXPT IS SET TO 0 AUTOMATICALLY.
33 E(ISTA)=0.,Q(ISTA+1),...E(NSAV-1),Q(NSAV),E(NSAV)=0.,
34 FORM A DIAGONAL OF THE QD-ARRAY.
35 INITIALIZATION OF BOUNDARY VALUES
36 E(ISTA)=0.
37 NRAN=NSAV-1
38 E(NRAN+1)=0.
C
C      TEST FOR LINEAR OR CONSTANT FACTOR
39 NRAN-ISTA IS DEGREE-1
40 IF(NRAN-ISTA)24,23,31
41 LINEAR FACTOR
42 Q(ISTA+1)=Q(ISTA+1)+EXPT

```



```

E(ISTA+1)=0.
      TEST FOR UNFACTORED COMMON DIVISOR
24  E(ISTA)=ESAV
    IF(IR-NSAV)60,60,25
      INITIALIZE QD-ALGORITHM FOR COMMON DIVISOR
25  ISTA=NSAV
    ESAV=E(ISTA)
    GOTO 10
      COMPUTATION OF ROOT PAIR
26  P=F+EXPT
      TEST FOR REALITY
    IF(C)27,28,28
      COMPLEX ROOT PAIR
27  Q(NRAN)=P
    C(NRAN+1)=P
    E(NRAN)=T
    E(NRAN+1)=-T
    GOTO 29
      REAL ROOT PAIR
28  Q(NRAN)=P-T
    C(NRAN+1)=P+T
    E(NRAN)=0.
      REDUCTION OF DEGREE BY 2 (DEFLATION)
29  NFAN=NRAN-2
    GOTO 22
      COMPUTATION OF REAL ROOT
30  C(NRAN+1)=EXPT+P
      REDUCTION OF DEGREE BY 1 (DEFLATION)
    NRAN=NRAN-1
    GOTO 22
      START QD-ITERATION
31  JBEG=ISTA+1
    JEND=NRAN-1
    TEPS=EPS
    TCELT=1.E-2
    KCUNT=KCUNT+1
    P=C(NRAN+1)
    R=ABS(E(NRAN))
32

```



```

C C      TEST FOR CONVERGENCE
C C      IF(R-TEPS)30,30,33
C C      S=ABS(E(JEND))
C C      IS THERE A REAL ROOT NEXT
C C      IF(S-R)38,38,34
C C      IS DISPLACEMENT SMALL ENOUGH
C C      IF(R-TDELT)36,35,35
C C      P=0.
C C      Q=P
C C      DO 37 J=JBEG,NRAN
C C      Q(J)=Q(J)+E(J)-E(J-1)-Q
C C      TEST FOR SMALL DIVISOR
C C      IF(ABS(Q(J))-POL(J))81,81,37
C C      E(J)=Q(J+1)*E(J)/Q(J)
C C      Q(NRAN+1)=-E(NRAN)+Q(NRAN+1)-Q
C C      GOTO 54
C C      CALCULATE DISPLACEMENT FOR DOUBLE ROOTS
C C      QUADRATIC EQUATION FOR DOUBLE ROOTS
C C      X*2-(Q(NRAN)+Q(NRAN+1))+E(NRAN))*X+Q(NRAN)*Q(NRAN+1)=0
C C      P=0.5*(Q(NRAN)+E(NRAN)+Q(NRAN+1))
C C      Q=P*P-Q(NRAN)*Q(NRAN+1)
C C      T=SQRT(ABS(Q))
C C      TEST FOR CONVERGENCE
C C      IF(S-TEPS)26,26,39
C C      ARE THERE COMPLEX ROOTS
C C      IF(C)43,40,40
C C      IF(P)42,41,41
C C      T=-T
C C      P=P+T
C C      R=S
C C      GOTO 34
C C      MODIFICATION FOR COMPLEX ROOTS
C C      IS DISPLACEMENT SMALL ENOUGH
C C      IF(S-TDELT)44,35,35
C C      INITIALIZATION
C C      Q=C(JBEG)+E(JBEG)-P
C C      TEST FOR SMALL DIVISOR
C C      IF(ABS(Q)-POL(JBEG))81,81,45

```

```

PRQD2150
PRQD2160
PRQD2170
PRQD2180
PRQD2190
PRQD2200
PRQD2210
PRQD2220
PRQD2230
PRQD2240
PRQD2250
PRQD2260
PRQD2270
PRQD2280
PRQD2290
PRQD2300
PRQD2310
PRQD2320
PRQD2330
PRQD2340
PRQD2350
PRQD2360
PRQD2370
PRQD2380
PRQD2390
PRQD2400
PRQD2410
PRQD2420
PRQD2430
PRQD2440
PRQD2450
PRQD2460
PRQD2470
PRQD2480
PRQD2490
PRQD2500
PRQD2510
PRQD2520
PRQD2530
PRQD2540
PRQD2550
PRQD2560
PRQD2570
PRQD2580
PRQD2590
PRQD2600
PRQD2610
PRQD2620

```



```

DC 75 I=1,IR
IF(C(I))72,71,72
71 O=ABS(PGL(I))
GOTO 73
72 C=ABS((POL(I)-C(I))/C(I))
73 IF(P-O)74,75,75
74 P=O
75 CCNTINUE
IF(P-TOL)77,76,76
76 IER=-1
77 C(IR+1)=P
E(IR+1)=0.
78 RETURN

C      ERROR RETURNS
C      ERROR RETURN FOR POLYNOMIALS OF DEGREE LESS THAN 1
C
79 IER=2
IR=0
RETURN

C      ERROR RETURN IF THERE EXISTS NO S-FRACTION
C
80 IER=4
IR=ISTA
GCTC 60

C      ERROR RETURN IN CASE OF INSTABLE QD-ALGORITHM
C
81 IER=3
GCTC 56
END

```

```

PRQD3590
PRQD3600
PRQD3610
PRQD3620
PRQD3630
PRQD3640
PRQD3650
PRQD3660
PRQD3670
PRQD3680
PRQD3690
PRQD3700
PRQD3710
PRQD3720
PRQD3730
PRQD3740
PRQD3750
PRQD3760
PRQD3770
PRQD3780
PRQD3790
PRQD3800
PRQD3810
PRQD3820
PRQD3830
PRQD3840
PRQD3850
PRQD3860
PRQD3870

```


BIBLIOGRAPHY

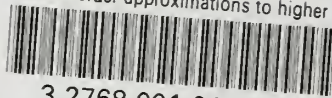
1. Hutton, M. F. and Friedland, B., "Routh Approximations of Reducing Order of Linear, Time-Invariant Systems," IEEE Transactions on Automatic Control, Vol. AC-20, pp 329-337.
2. Cantalapiedra, F., Low-Order Models for Dynamic Systems, MS Thesis, N. P. G. School, Monterey, California, June, 1972.
3. "Subroutine PRQD," W. R. Church Computer Facility Subroutine Library, N. P. G. School, Monterey, California.
4. Ward, J. R., "Subroutine INTEGL," W. R. Church Computer Facility Subroutine Library, N. P. G. School, Monterey, California.
5. Davison, E. J., "A Method for Simplifying Linear Dynamic Systems," IEEE Transactions on Automatic Control, Vol. AC-11, 1966.
6. Lal, M. and Van Valkenburg, M. E., "A Model-Reduction Method for Large Linear Systems," Ninth Annual Asilomar Conference on Circuits, Systems, and Computers, Nov, 1975.
7. Computer and Electrical Engineering Report, Vol. 2, pp 117-123, Computer Determination of Low-Order Models for High Order Systems, by G. J. Thaler, 22 May, 1974.

INITIAL DISTRIBUTION LIST

	Copies
1. Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2. Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3. Department Chairman, Code 52 Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	2
4. Professor A. Gerba, Jr. Code 52 Gz Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	2
5. Professor G. J. Thaler, Code 52 Tr Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
6. LT Jerry D. Thompson 62 West South Street Mooreville, Indiana 46158	1

thesT43525

Reduced order approximations to higher o



3 2768 001 01099 4

DUDLEY KNOX LIBRARY